

Garage Management System



COLLEGE NAME: NAVARASAM ARTS AND SCIENCE COLLEGE FOR WOMEN

COLLEGE CODE: asbru3a

TEAM ID: NM2025TMID27240

TEAM LEADER NAME: KARTHIYAYINI.TS

EMAIL: karthiyayinisivakumar2006@gmail.com

TEAM MEMBER1: SANTHIYA.R

EMAIL: rangarajansanthiya900@gmail.com

TEAM MEMBER2: GAYATHRI.M

EMAIL: gayu31284@gmail.com

TEAM MEMBER3: SOWMIYA.C

EMAIL: sowmiyachenniyappan@gmail.com

GARAGE MANAGEMENT SYSTEM

PROJECT OVERVIEW:

The Garage Management System project uses Salesforce CRM to build an application that manages the operations of vehicle service garages. The system connects customers, mechanics, inventory managers, and administrators to streamline service workflows. With the help of custom objects, automation, flows, and dashboards, the application handles customer details, vehicle records, service scheduling, spare parts tracking, and billing in a centralized platform. The process becomes efficient, transparent, and paperless, while customers receive timely updates and better service. This project demonstrates how technology can be applied to optimize garage management and improve customer satisfaction.

OBJECTIVES:

- To maintain accurate records of customers and their vehicles.
- To automate service scheduling, reminders, and job assignments.
- To track spare parts inventory and procurement.
- To generate invoices, estimates, and payment records efficiently.
- To improve customer engagement using Salesforce features (emails, SMS, service updates).
- To provide dashboards and reports for performance analysis and decision-making.

STUDENTS OUTCOMES

- **Hands-on Experience with Garage Management Automation:** Students gain practical skills in configuring Salesforce objects, automating service workflows, and managing real-time vehicle and inventory tracking.
- **Understanding of Project Lifecycle in Salesforce CRM:** Students learn the complete end-to-end process from requirement gathering to deployment, enhancing their ability to execute real-world Salesforce projects.
- **Enhanced Analytical and Problem-Solving Skills:** Students develop the ability to identify operational challenges in garage management, design solutions, and troubleshoot issues effectively.
- **Improved Collaboration Skills:** Students gain experience in teamwork, coordinating tasks such as requirement gathering, development, and testing.
- **Industry-Relevant Exposure:** Students get exposure to real-world use cases of Salesforce CRM in service management projects, preparing them for future career opportunities.

SYSTEM REQUIREMENTS

Hardware Requirements:

- Computer with minimum 4 GB RAM, Dual-core processor
- Stable internet connection

Software Requirements:

- Salesforce Developer Edition Org
- Modern Web Browser (e.g., Google Chrome, Firefox, Edge)

PURPOSE: To streamline and automate garage operations using Salesforce CRM for efficient service management and improved customer satisfaction.

PHASES 1

REQUIREMENTS ANALYSIS & PLANNING: -

The project requires a cloud-based Garage Management System to handle customer details, vehicle records, service scheduling, inventory, and billing. Functional and non-functional needs were identified, and modules were planned in Salesforce with clear milestones for design, development, testing, and deployment.

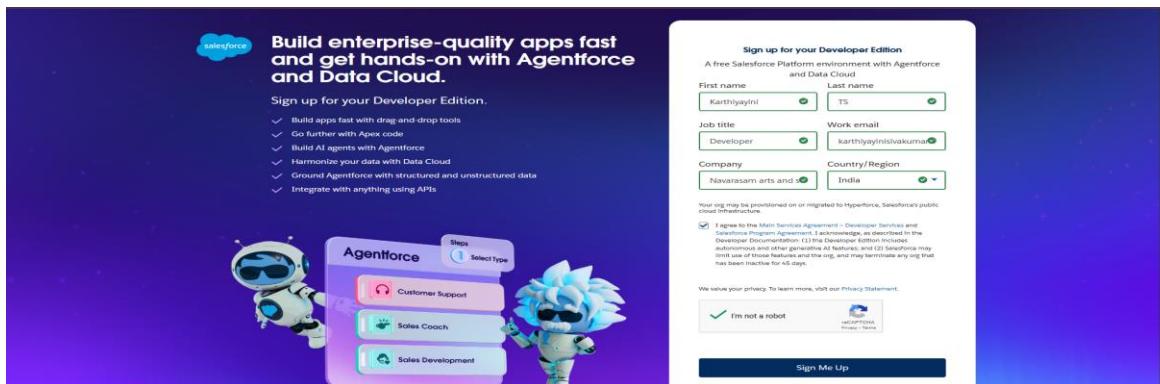
PHASE 2

DEVELOPMENT PHASE

MILESTONE 1: Salesforce developer account creation

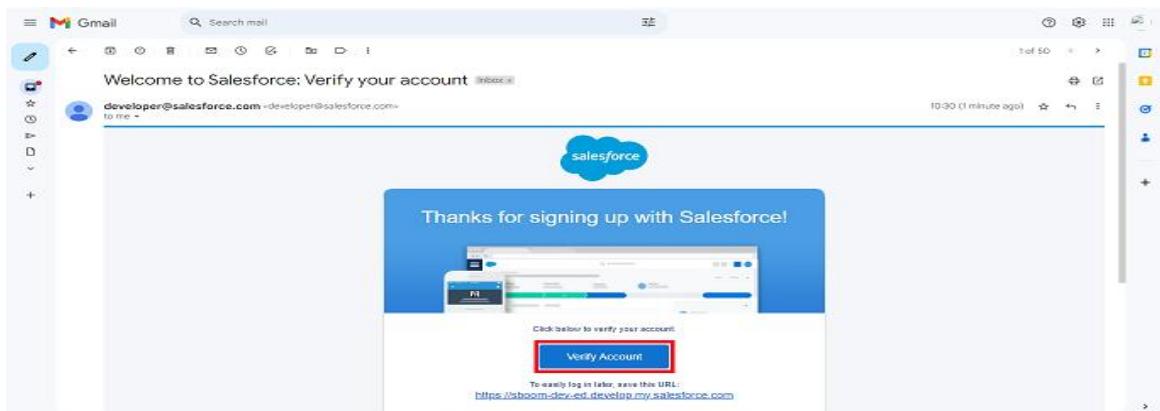
ACTIVITY 1: Creating Developer Account:

By using this URL: <https://www.salesforce.com/form/developer-signup/?d=pb>



ACTIVITY 2: Account Activation

Go to the inbox of the email that you used while signing up. Click on the verify account to activate your account. The email may take 5-10mins.



Give a password and answer a security question and click on change password.

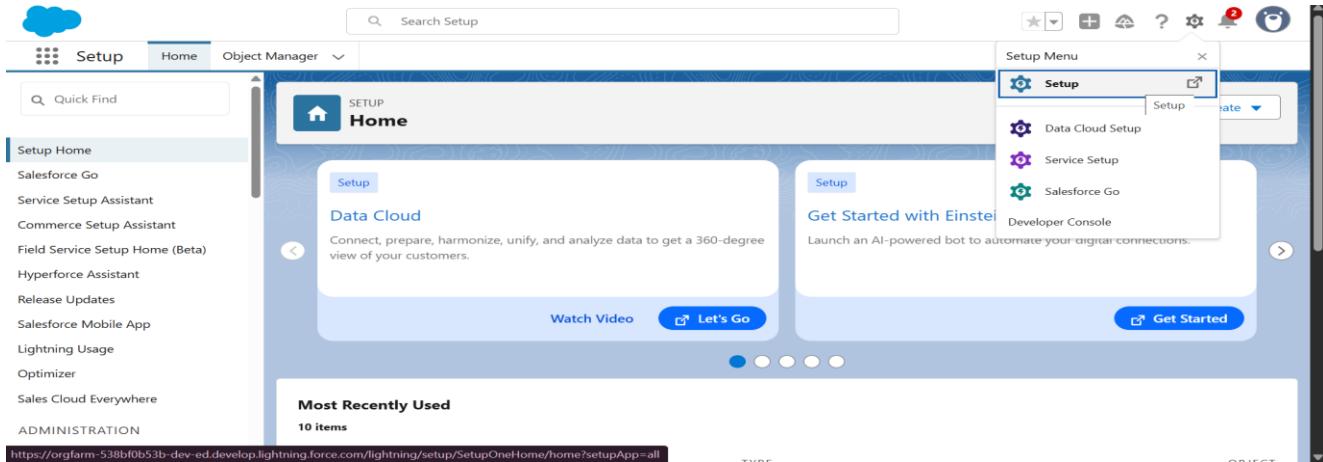
The screenshot shows the 'Change Your Password' page. At the top, it says 'Enter a new password for lead@sb.com. Make sure to include at least:'. Below this are three requirements: '8 characters', '1 letter', and '1 number'. A red box highlights the 'New Password' field, which contains 'Gooooo'. Another red box highlights the 'Confirm New Password' field, which contains 'Match'. A third red box highlights the 'Answer' field, which contains 'asdfghijkl'. At the bottom is a blue 'Change Password' button.

Then you will redirect to your salesforce setup page.

The screenshot shows the Salesforce Setup Home page. The left sidebar includes links like 'Setup Home', 'Salesforce Go', 'Service Setup Assistant', 'Commerce Setup Assistant', 'Field Service Setup Home (Beta)', 'Hyperforce Assistant', 'Release Updates', 'Salesforce Mobile App', 'Lightning Usage', 'Optimizer', 'Sales Cloud Everywhere', and 'ADMINISTRATION'. The main content area features a 'Data Cloud' section with 'Watch Video' and 'Let's Go' buttons, and a 'Get Started with Einstein Bots' section with a 'Get Started' button. Below these are sections for 'Most Recently Used' items.

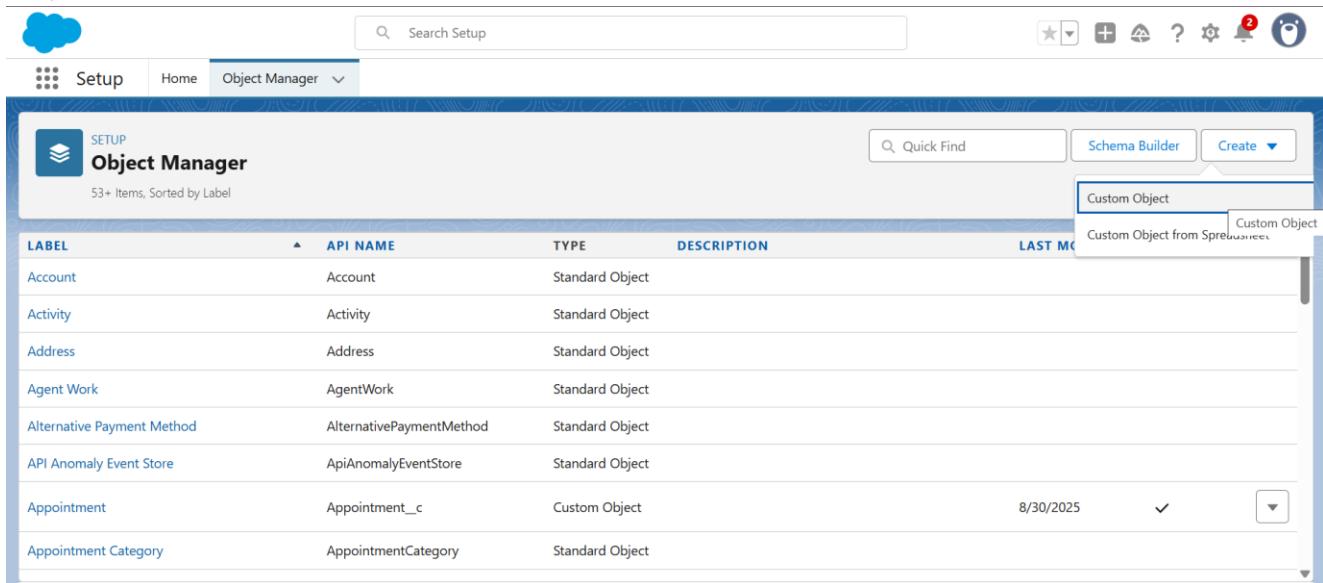
MILESTONE 2: Object

To Navigate to Setup page:
Click on gear icon? click setup.



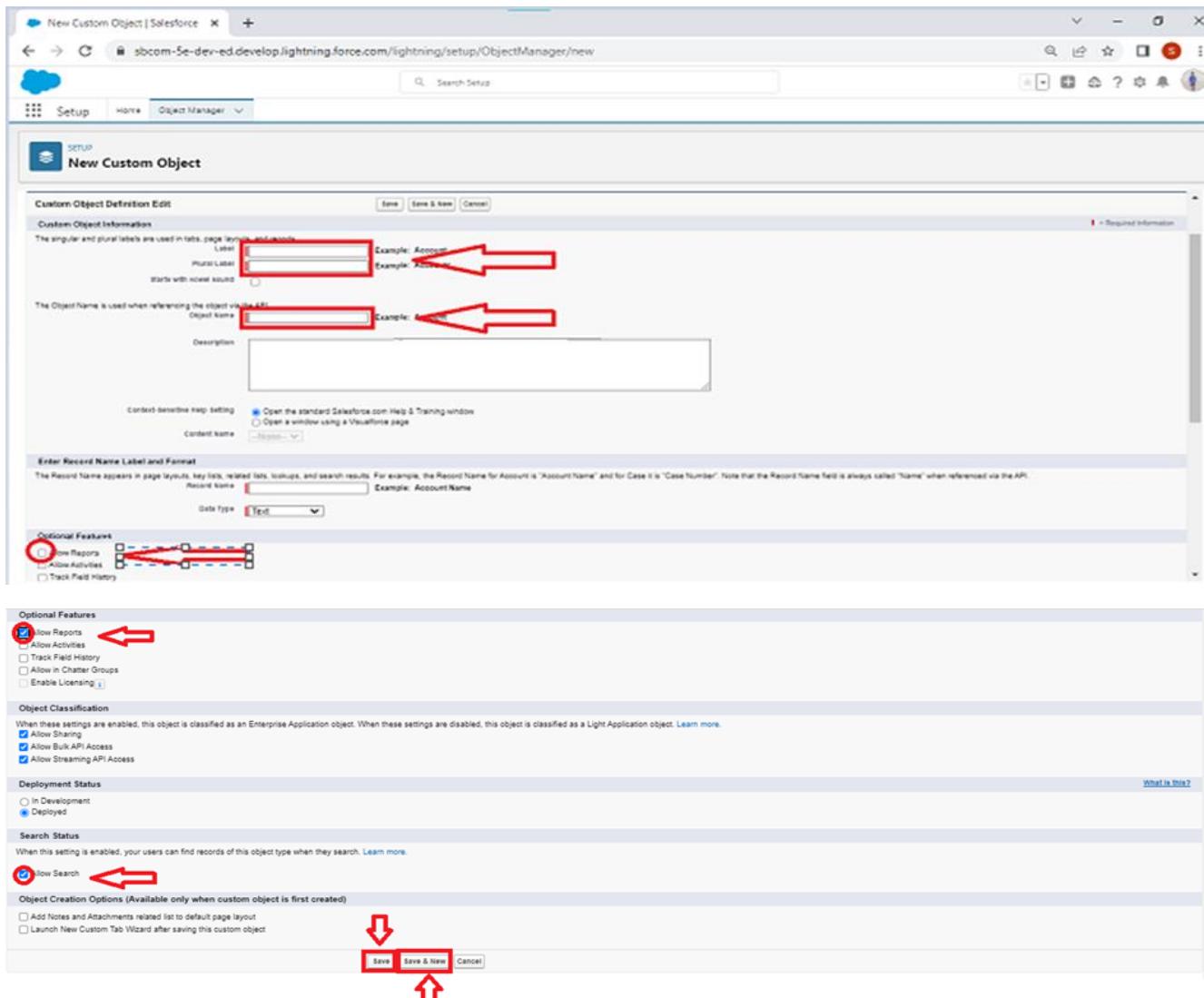
To create an object:

From the setup page? Click on Object Manager? Click on Create? Click on Custom Object.



On Custom object defining page:

Enter the label name, plural label name, click on Allow reports, Allow search.



Click on save

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Customer Details
2. Plural label name >> Customer Details
3. Enter Record Name Label and Format
 - Record Name >> Customer Name
 - Data Type >> Text
2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Appointment
2. Plural label name >> Appointments
3. Enter Record Name Label and Format
 - Record Name >> Appointment Name
 - Data Type >> Auto Number
 - Display Format >> app- {000}
 - Starting number >> 1
2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

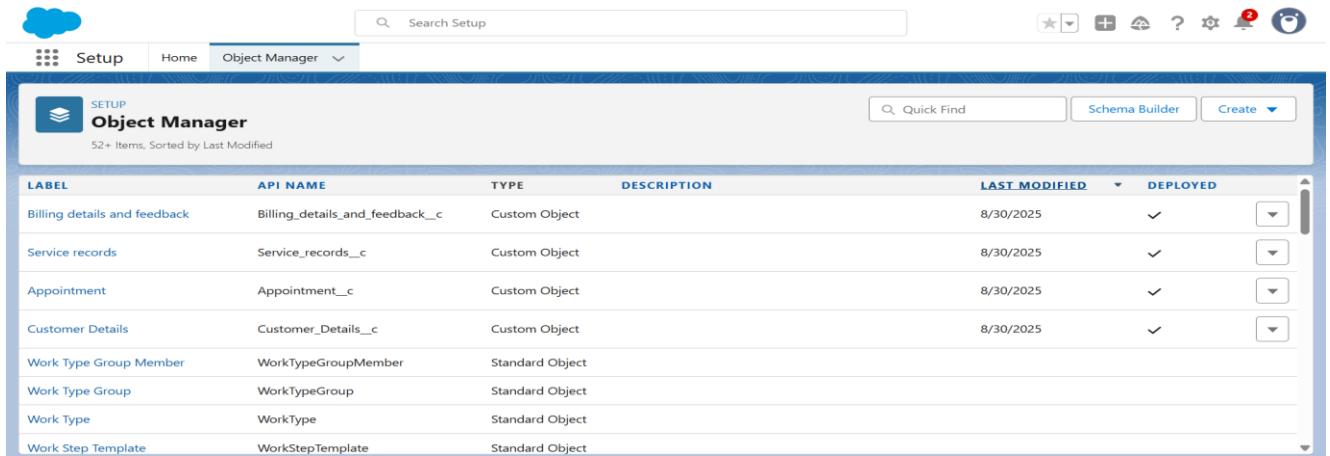
To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Appointment
2. Plural label name >> Appointments
3. Enter Record Name Label and Format
 - Record Name >> Appointment Name
 - Data Type >> Auto Number
 - Display Format >> app- {000}
 - Starting number >> 1
2. Click on Allow reports and Track Field History,
3. Allow search >> Save.

To create an object:

1. From the setup page >> Click on Object Manager >> Click on Create >> Click on Custom Object.
1. Enter the label name >> Billing details and feedback
2. Plural label name >> Billing details and feedback
3. Enter Record Name Label and Format
 - Record Name >> Billing details and feedback Name
 - Data Type >> Auto Number

- Display Format >> bill-{000}
 - Starting number >> 1
2. Click on Allow reports and Track Field History,
 3. Allow search >> Save.



The screenshot shows the Salesforce Object Manager interface. At the top, there's a navigation bar with icons for Setup, Home, and Object Manager. A search bar says "Search Setup". Below the header, the title "Object Manager" is displayed with a subtitle "52+ Items, Sorted by Last Modified". There are buttons for "Quick Find", "Schema Builder", and "Create". The main area is a table with columns: Label, API Name, Type, Description, Last Modified, and Deployed. The table lists several objects:

Label	API Name	Type	Description	Last Modified	Deployed
Billing details and feedback	Billing_details_and_feedback__c	Custom Object		8/30/2025	✓
Service records	Service_records__c	Custom Object		8/30/2025	✓
Appointment	Appointment__c	Custom Object		8/30/2025	✓
Customer Details	Customer_Details__c	Custom Object		8/30/2025	✓
Work Type Group Member	WorkTypeGroupMember	Standard Object			
Work Type Group	WorkTypeGroup	Standard Object			
Work Type	WorkType	Standard Object			
Work Step Template	WorkStepTemplate	Standard Object			

These are created Objects.

MILESTONE 3: Tabs

ACTIVITY 1: Creating custom tabs

To create a Tab:(Customer Details)

1. Go to setup page >> type Tabs in Quick Find bar >> click on tabs >> New (under custom object tab) Select Object(Customer Details) >> Select the tab style >> Next (Add to profiles page) keep it as default >> Next (Add to Custom App) uncheck the include tab .

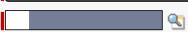
Screenshot of the Salesforce Setup interface showing the 'Tabs' section. The 'Custom Tabs' tab is selected. A red box highlights the 'Custom Tabs' link in the left sidebar. Another red box highlights the 'New' button in the top right corner of the main list area.

New Custom Object Tab

Step 1. Enter the Details Step 1 of 3

Choose the custom object for this new custom tab. Fill in other details.

Select an existing custom object or [create a new custom object now](#).

Object	<input type="text" value="Customer Details"/>
Tab Style	

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

Splash Page Custom Link	--None--
-------------------------	----------

Enter a short description.

Description	<input type="text"/>
-------------	----------------------

[Next](#) [Cancel](#)

Tab Style Selector

Create your own style

Hide styles which are used on other tabs

Airplane	Alarm clock	Apple	Balls	Bank	Bell[1]	Big top
Boat	Books[1]	Bottle	Box	Bridge	Building	Building Block
Caduceus	Camera	Can	Car	Castle	Cell phone	Credit card
Chalkboard	Chess piece	Chip	Circle	Compass	Computer[1]	Fan
CRT TV	Cup	Desk	Diamond	Dice	Factory	Hands
Flag	Form	Gears	Globe	Guitar	Hammer	Hot Air Balloon
Handsaw	Headset	Heart	Helicopter	Hexagon	Highway Sign	Lightning
Insect	IP Phone	Jewel	Keys	Laptop[1]	Leaf	Motorcycle
Locked	Mail	Map	Measuring Tape	Microphone	Moon	Presenter
Musical Note	PDA	Pencil	People	Phone	Postage	Saxophone
Radar dish	Real Estate Sign	Red Cross	Sack	Safe	Sailboat	Star
Scales	Shield	Stopwatch	Shopping Cart	Square	Stack of Cash	Ticket
Stethoscope	Street Sign	Treasure chest	Sun	Telescope	Thermometer	Umbrella
Train	Triangle	Wrench	Trophy	Truck	TV Widescreen	

New Custom Object Tab

Step 2. Add to Profiles Step 2 of 3

Choose the user profiles for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each profile.

Apply one tab visibility to all profiles (Default On) Apply a different tab visibility for each profile

Profile	Tab Visibility
Analytics Cloud Integration User	Default On
Analytics Cloud Security User	Default On
Anypoint Integration	Default On
Authenticated Website	Default On
Authenticated Website	Default On
B2B Reordering Portal Buyer Profile	Default On
Contract Manager	Default On
Cross Org Data Proxy User	Default On
Custom: Marketing Profile	Default On
Custom: Sales Profile	Default On
Custom: Support Profile	Default On

Step 3. Add to Custom Apps Step 3 of 3

Choose the custom apps for which the new custom tab will be available. You may also examine or alter the visibility of tabs from the detail and edit pages of each Custom App.

include Tab

Custom App	Include Tab
Platform (standard__Platform)	<input checked="" type="checkbox"/>
Sales (standard__Sales)	<input checked="" type="checkbox"/>
Service (standard__Service)	<input checked="" type="checkbox"/>
Marketing CRM Classic (standard__Marketing)	<input checked="" type="checkbox"/>
Sample Console (standard__ServiceConsole)	<input checked="" type="checkbox"/>
High Volume Customer Portal User	<input checked="" type="checkbox"/>
Authenticated Website User	<input checked="" type="checkbox"/>
App Launcher (standard__AppLauncher)	<input checked="" type="checkbox"/>
Community (standard__Community)	<input checked="" type="checkbox"/>
Site.com (standard__Sites)	<input checked="" type="checkbox"/>
Salesforce Chatter (standard__Chatter)	<input checked="" type="checkbox"/>
Content (standard__Content)	<input checked="" type="checkbox"/>
Analytics Studio (standard__Insights)	<input checked="" type="checkbox"/>
Sales Console (standard__LightningSalesConsole)	<input checked="" type="checkbox"/>

Setup Home Object Manager

Tabs

Append tab to users' existing personal customizations

Object	Tab Visibility
Service Console (standard__LightningService)	<input checked="" type="checkbox"/>
Sales (standard__LightningSales)	<input checked="" type="checkbox"/>
Lightning Usage App (standard__LightningInstrumentation)	<input checked="" type="checkbox"/>
Digital Experiences (standard__SalesforceCMS)	<input checked="" type="checkbox"/>
Queue Management (standard__QueueManagement)	<input checked="" type="checkbox"/>
Data Manager (standard__DataManager)	<input checked="" type="checkbox"/>
Subscription Management (standard__RevenueCloudConsole)	<input checked="" type="checkbox"/>
Sales Cloud Mobile (standard__SalesCloudMobile)	<input checked="" type="checkbox"/>
Bolt Solutions (standard__LightningBolt)	<input checked="" type="checkbox"/>
Data Cloud (standard__Audience360)	<input checked="" type="checkbox"/>
Approvals (standard__Approvals)	<input checked="" type="checkbox"/>
My Service Journey (standard__MSJApp)	<input checked="" type="checkbox"/>
Salesforce Scheduler Setup (standard__LightningScheduler)	<input checked="" type="checkbox"/>
Automation (standard__FlowsApp)	<input checked="" type="checkbox"/>
Garage Management Application (Garage_Management_Application)	<input checked="" type="checkbox"/>

Previous Save Cancel

ACTIVITY 2: Creating remaining tabs

1. Now create the Tabs for the remaining Objects, they are “ Appointments, Service records,Billing details and feedback”.

2. Follow the same steps as mentioned in Activity -1.

Finally new custom tabs are created Successfully

MILESTONE 4: The Lightning App

ACTIVITY 1: Creating a Lightning App

To create a lightning app page:

1. Go to setup page >> search “app manager” in quick find >> select “app manager” >> click on new lightning App.

The screenshot shows the Salesforce App Manager interface. At the top, there are three red boxes highlighting the search bar ('Q app manager'), the 'App Manager' tab, and the 'Clone (Preview Beta)' button. A red arrow points to the 'New Lightning App' button on the right side of the header. Below the header, there is a message about cloning existing apps. The main area displays a table of existing apps, with one row selected. The table has columns for App Name, Developer Name, Description, Last Modified, App Type, and more. The selected row shows 'All Tabs' as the app name, 'AirTable' as the developer name, and 'Build CRM Analytics dashboards and apps' as the description.

2. Fill the app name in app details as Garage Management Application >> Next >> (App option page) keep it as default >> Next >> (Utility Items) keep it as default >> Next.

New Lightning App

App Details & Branding

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details

App Name *

Developer Name

Description

App Branding

Image *

Primary Color Hex Value *

Org Theme Options Use the app's image and color instead of the org's custom theme

App Launcher Preview

Next

New Lightning App

App Options

Navigation and Form Factor *

- Navigation Style** Standard navigation Console navigation
- Supported Form Factors** Desktop and phone Desktop Phone

Setup and Personalization *

Setup Experience Setup (full set of Setup options) Service Setup Data Cloud Setup

App Personalization Settings

- Disable end user personalization of nav items in this app
- Disable temporary tabs for items outside of

Next

New Lightning App

Utility Items (Desktop Only)

Give your users quick access to productivity tools and add background utility items to your app.

Add Utility Item Default

The utility bar is a fixed footer that opens components in docked panels. Available only when the app is viewed in Lightning Experience on a desktop.

Next

3.To Add Navigation Items:

New Lightning App

Navigation Items

Choose the items to include in the app, and arrange the order in which they appear. Users can personalize the navigation to add or move items, but users can't remove or rename the items that you add. Some navigation items are available only for phone or only for desktop. These items are dropped from the navigation bar when the app is viewed in a format that the item doesn't support.

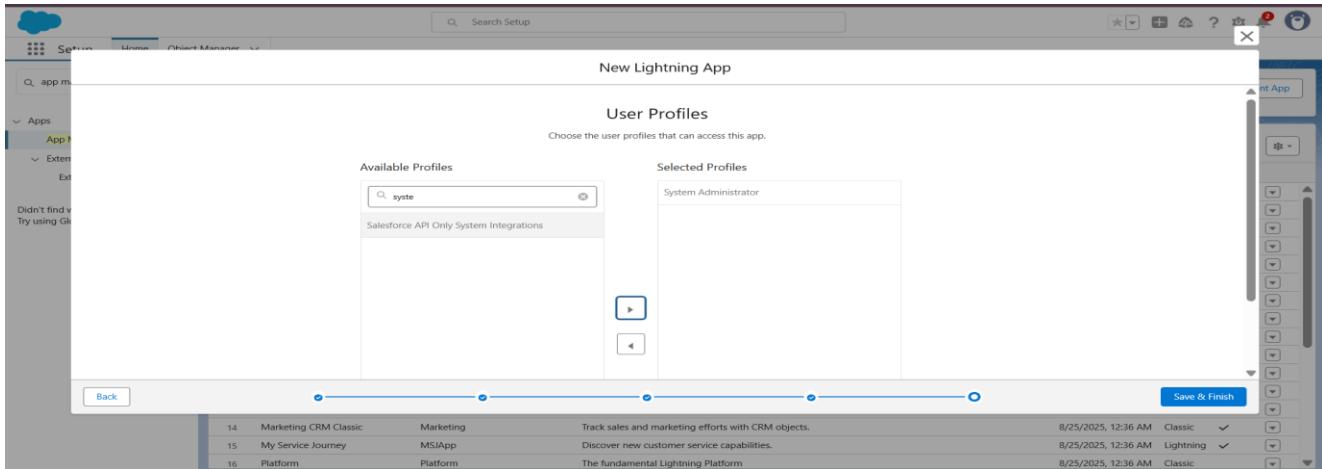
Available Items	Selected Items	
<input type="text" value="dash"/> <input type="button" value="Create"/> <input style="border: none; border-bottom: 1px solid #ccc; padding: 2px 10px;" type="button" value="..."/>	<div style="border: 1px solid #ccc; padding: 5px; display: inline-block;"> Customer Details Appointments Service records Billing details and feedback Reports Dashboards </div>	<input type="button" value="Up"/> <input type="button" value="Down"/>

Next

4.Select the items (Customer Details,Appointments, Service records, Billing details and feedback, Reports and Dashboards) from the search bar and move it using

the arrow button >> Next.

5.To Add User Profiles :



Search profiles (System administrator) in the search bar >> click on the arrow button >> save & finish.

The image contains three screenshots illustrating the creation and use of a custom app:

- Screenshot 1: App Launcher**
Shows the 'App Launcher' interface with a search bar containing 'Gar'. A sidebar lists various apps like 'Hyperforce Assistant', 'Release Updates', and 'Salesforce Mobile App'. The 'Lightning Usage' tab is selected.
- Screenshot 2: Lightning Experience App Manager**
Shows the 'Lightning Experience App Manager' screen. It lists various apps in a table:

App Name ↑	Developer Name	Description	Last Modified ...	Ap... ▾	Vi... ▾
1 Tabs	AllTabSet	Build CRM Analytics dashboard	8/25/2025, 12:36 AM	Classic	▼
2 Analytics Studio	Insights	Build CRM Analytics dashboard	8/25/2025, 12:36 AM	Classic	▼
3 App Launcher	AppLauncher	App Launcher tabs	8/25/2025, 12:36 AM	Classic	▼
4 Approvals	Approvals	Manage approvals and approv...	8/25/2025, 12:36 AM	Lightning	▼
5 Automation	FlowsApp	Automate business processes ...	8/25/2025, 12:44 AM	Lightning	▼
6 Bolt Solutions	LightningBolt	Discover and manage business...	8/25/2025, 12:36 AM	Lightning	▼
7 Community	Community	Salesforce CRM Communities	8/25/2025, 12:36 AM	Classic	▼
8 Content	Content	Salesforce CRM Content	8/25/2025, 12:36 AM	Classic	▼
9 Data Cloud	Audience360	Build a thorough and complet...	8/25/2025, 12:36 AM	Lightning	▼
- Screenshot 3: Garage Management App**
Shows the 'Customer Details' tab of the 'Garage Management' app. It displays a 'Recently Viewed' section with 10 items. A search bar at the bottom has the placeholder 'Your search term must have 2 or more characters.'

New Lightning app was created successfully.

MILESTONE 5: Fields

ACTIVITY 1: Creation of fields for the Customer Details object

1. To create fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Customer Details) in search bar >> click on the object.

The screenshot shows the Salesforce Object Manager interface. A red box highlights the row for 'Customer Details' in the list. The columns shown are Label, API Name, Type, Description, Last Modified, and Deployed. The 'Customer Details' row has 'Customer_Details__c' in the API Name column and 'Custom Object' in the Type column. The 'Last Modified' column shows '05/10/2023'.

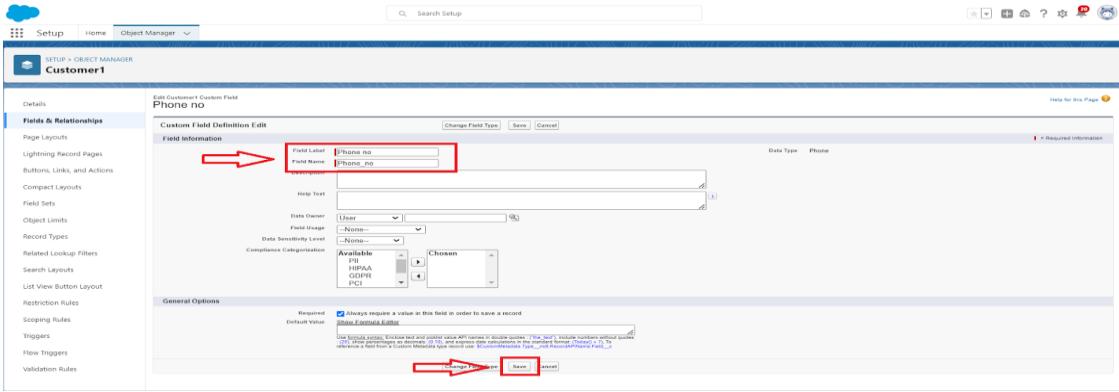
2. Now click on “Fields & Relationships” >> New

The screenshot shows the 'Fields & Relationships' section of the Object Manager for the 'Customer' object. A red box highlights the 'Customer' label in the top navigation bar. Another red box highlights the 'Fields & Relationships' tab. A third red box highlights the 'New' button in the top right corner of the list view. The table lists various fields like 'Created By', 'current_Status', etc., with their field names, data types, and controlling fields.

3. Select Data Type as a “Phone”

The screenshot shows the 'Field Types' dropdown menu for selecting a data type. A red box highlights the 'Phone' option under the 'Text' category. A detailed description of the 'Phone' type is visible at the bottom of the list.

4. Click on next.



5. Fill the Above as following:

- Field Label: Phone number
- Field Name: gets auto generated
- Click on Next >> Next >> Save and new.

Note: Follow the above steps for the remaining field for the same object.

2. To create other fields in an object:

1. Go to setup >> click on Object Manager >> type object name (Customer Details) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select Data type as a “Email” and Click on Next
4. Fill the Above as following:
 - Field Label: Gmail
 - Field Name: gets auto generated
 - Click on Next >> Next >> Save and new.

ACTIVITY 2: Creation of Lookup Fields

1. Creation of Lookup Field on Appointment Object :

1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select “Look-up relationship” as data type and click Next.
4. Select the related object “ Customer Details” and click next.
5. Next >> Next >> Save.

Note: Make sure you complete Activity 4 Before continuing.

2.Creation of Lookup Field on Service records Object :

1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New
3. Select “Look-up relationship” as data type and click Next.
4. Select the related object “ Appointment ” and click next.
5. Make it a required field so click on Required.
6. Scroll down for Lookup Filter and click on Show filter settings.
7. Now add the filter criteria.
8. Field : Appointment: Appointment Date >> Operator : less than >> select field >> Appointment: Created Date
9. Filter type should be Required.
10. Error Message : Value does not match the criteria.
11. Enable the filter by click on Active.
12. Next >> Next >> Save.

3.Creation of Lookup Field on Billing details and feedback Object :

1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Look-up relationship” as data type and click Next.
4. Select the related object “ Service records ” and click next.
5. Next >> Next >> Save & new.

ACTIVITY 3: Creation of Checkbox Fields

1. Creation of Checkbox Field on Appointment Object:

1. Go to setup >> click on Object Manager >> type object name (Appointment) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.

3. Select “Check box” as data type and click Next.
4. Give the Field Label: Maintenance service
5. Field Name: is auto populated
6. Default value: unchecked
7. Click on next >> next >> save.

2.Creation of Another Checkbox Field on Appointment Object :

1. Repeat the steps from 1 to 3.
2. Give the Field Label: Repairs
3. Field Name: is auto populated
4. Default value: unchecked
5. Click on next >> next >> save.
6. Follow the same and create another checkbox with given names
7. Give the Field Label: Replacement Parts
8. Field Name: is auto populated
9. Default value: unchecked
10. Click on next >> next >> save.

3.Creation of Checkbox Field on Service records Object:

1. Go to setup >> click on Object Manager >> type object name (Service records) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Check box” as data type and click Next.
4. Give the Field Label: Quality Check Status
5. Field Name: is auto populated
6. Default value: unchecked
7. Click on next >> next >> save.

ACTIVITY 4: Creation of Date Fields

1.Creation of Date Field on Appointment Object:

1. Go to setup >> click on Object Manager >> type object name (Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Date” as data type and click Next.

4. Give the Field Label: Appointment Date
5. Field Nme : is auto populated
6. Make it as a Required field by click on the Required option.
7. Click on next >> next >> save.

ACTIVITY 5: Creation of Currency Fields

1.Creation of Currency Field on Appointment Object:

1. Go to setup >> click on Object Manager >> type object name (Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Currency” as data type and click Next.
4. Give the Field Label: Service Amount
5. Field Name: is auto populated
6. Click on next
7. Give read only for all the profiles in field level security for profile.
8. Click on next >> save.

2.Creation of Currency Field on Billing details and feedback Object

Follow the same steps as mentioned above in Billing details and feedback Object

1. Change the label name as mentioned.
2. Give the Field Label: Payment Paid
3. Field Name : is auto populated

ACTIVITY 6: Creation of Text Fields

1.Creation of Text Field on Appointment Object:

1. Go to setup >> click on Object Manager >> type object name(Appointment) in the search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “Text” as data type and click Next.
4. Give the Field Label : Vehicle number plate
5. Field Name : is auto populated
6. Length : 10

7. Make field as Required and Unique.
8. Click on next >> next >> save.

2.Creation of Text Fields in Billing details and feedback object :

1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Now click on “Fields & Relationships” >> New.
3. Select “text” as data type and click Next.
4. Give the Field Label : Rating for service
5. Field Name : is auto populated
6. Length : 1
7. Make field as Required.
8. Click on next >> next >> save

ACTIVITY 7: Creation of Picklist Fields

1.Creation of Picklist Fields in Service records object :

1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Picklist” and click Next.
4. Enter Field Label as “Service Status”, under values select “Enter values, with each value separated by a new line” and enter values as shown below.
5. The values are: Started, Completed.
6. Click Next.
7. Next >> Next >> Save.

2. Creation of Picklist Fields in Billing details and feedback object :

1. Go to setup >> click on Object Manager >> type object name(Billing details and feedback) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Picklist” and click Next.

4. Enter Field Label as “Payment Status”, under values select “Enter values, with each value separated by a new line” and enter values as shown below.
5. The values are: Pending, Completed.
6. Click Next.
7. Next >> Next >> Save.

ACTIVITY 8: Creation of Formula Field in Service records Object

1. Go to setup >> click on Object Manager >> type object name(Service records) in search bar >> click on the object.
2. Click on fields & relationship >> click on New.
3. Select Data type as “Formula” and click Next.
4. Give Field Label and Field Name as “service date” and select formula return type as “Date” and click next.

Step 2. Choose output type

Field Label Field Name [Upload]

Auto add to custom report type Add this field to existing custom report types that contain this entity [i]

Formula Return Type

None Selected Select one of the data types below.

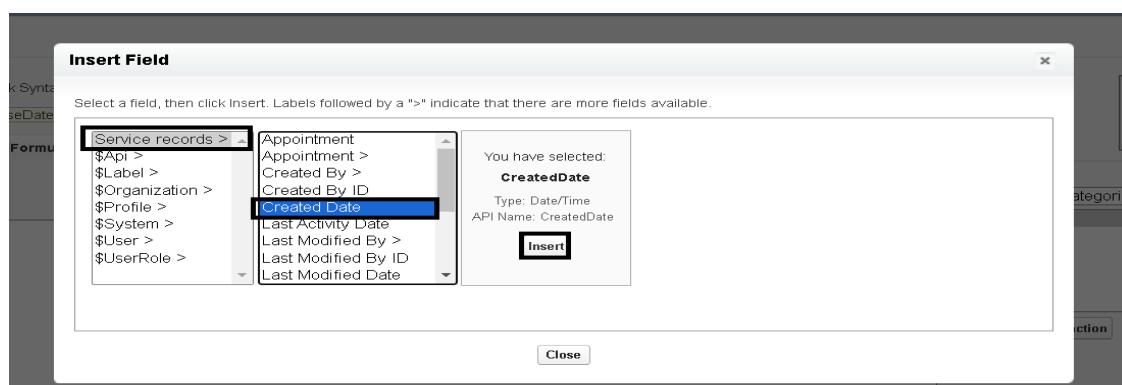
Checkbox Calculate a boolean value
Example: `[TODAY() > CloseDate]`

Currency Calculate a dollar or other currency amount and automatically format the field as a currency amount.
Example: `Gross Margin = Amount - Cost__c`

Date Calculate a date, for example, by adding or subtracting days to other dates.
Example: `Reminder Date = CloseDate - 7`

Date/Time Calculate a date/time, for example, by adding a number of hours or days to another date/time.
Example: `Next = NOW() + 1`

5. Insert field formula should be : CreatedDate



Step 3. Enter formula

Enter your formula and click Check Syntax to check for errors. Click the Advanced Formula subtab to use additional fields, operators, and functions.

Example: `Reminder Date = CloseDate - 7` [More Examples...](#)

[Simple Formula](#) [Advanced Formula](#)

[Insert Field](#) [Insert Operator](#)

`service_date (Date) =` [CreatedDate](#) 

[Functions](#) [All Function Categories](#)

- ABS
- ACOS
- ADDMONTHS
- AND
- ASCII

Quick Tips

- Getting Started
- Operators & Functions



6. click “Check Syntax” .
7. Click next >> next >> Save.

Setup > Object Manager **Customer Details**

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Text(80)		
Gmail	Gmail_c	Email		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Phone number	Phone_number_c	Phone		

Setup > Object Manager **Appointment**

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment Date	Appointment_Date_c	Date		
Appointment Name	Name	Auto Number		
Created By	CreatedById	Lookup(User)		
Customer Details	Customer_Details_c	Lookup(Customer Details)		
Last Modified By	LastModifiedById	Lookup(User)		
Maintenance service	Maintenance_service_c	Checkbox		
Owner	OwnerId	Lookup(User,Group)		
Repairs	Repairs_c	Checkbox		
Replacement Parts	Replacement_Parts_c	Checkbox		
Service Amount	Service_Amount_c	Currency(18, 0)		
Vehicle number plate	Vehicle_number_plate_c	Text(10) (Unique Case Insensitive)		

Setup > Object Manager **Service records**

Fields & Relationships

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Appointment	Appointment_c	Lookup(Appointment)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
Quality Check Status	Quality_Check_Status_c	Checkbox		
service date	service_date_c	Formula (Date)		
Service records Name	Name	Auto Number		
Service Status	Service_Status_c	Picklist		

NOTE: Thus we have successfully created all the fields which instructed to create on the respective object .

MILESTONE 6: Validation Rule

ACTIVITY 1: To create a validation rule to an Appointment Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Appointment object.
2. Click on the validation rule >> click New.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main content area is titled 'SETUP > OBJECT MANAGER Appointment'. On the left, there's a sidebar with various options like 'Related Lookup Filters', 'Search Layouts', 'List View Button Layout', etc., with 'Validation Rules' being the active tab. The main table lists one validation rule:

RULE NAME	ERROR LOCATION	ERROR MESSAGE	ACTIVE	MODIFIED BY
vehicle	Vehicle number plate	Please enter valid number	✓	Karthiyayini TS, 8/30/2025, 6:17 AM

3. Enter the Rule name as “Vehicle”.
4. Insert the Error Condition Formula as :-

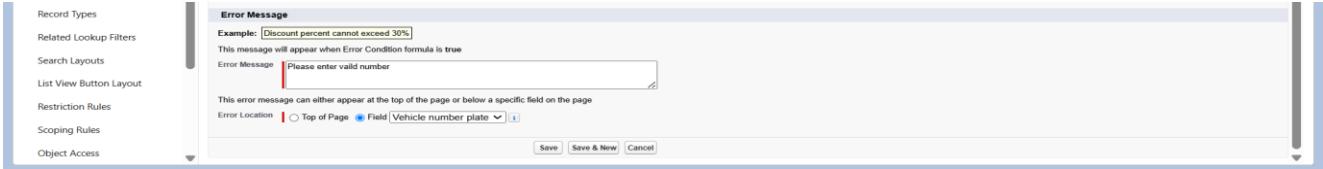
NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}"))

The screenshot shows the 'Appointment Validation Rule' edit screen. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. The main content area is titled 'SETUP > OBJECT MANAGER Appointment'. On the left, there's a sidebar with various options like 'Details', 'Fields & Relationships', etc. The right side shows the 'Appointment Validation Rule' configuration. The 'Validation Rule Edit' form has the following details:

- Rule Name: vehicle
- Active: checked
- Description: (empty)
- Error Condition Formula: NOT(REGEX(Vehicle_number_plate__c , "[A-Z]{2}[0-9]{2}[A-Z]{2}[0-9]{4}")))

The formula is highlighted in red, indicating an error or warning.

5. Enter the Error Message as “Please enter valid number”, select the Error location as Field and select the field as “Vehicle number plate”, and click Save.



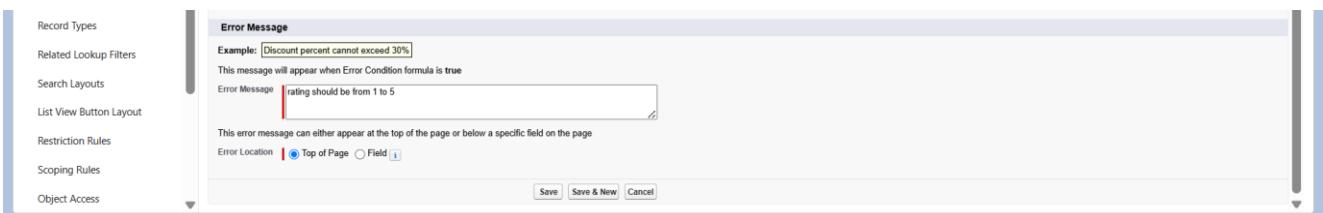
ACTIVITY 2: To create a validation rule to an Billing details & feedback Object

1. Go to the setup page >> click on object manager >> From drop down click edit for Billing details and feedback object.
2. Click on the validation rule >> click New.
3. Enter the Rule name as “ rating_should_be_less_than_5”.
4. Insert the Error Condition Formula as :-

NOT(REGEX(Rating_for_service_c , "[1-5]{1}"))

The screenshot shows the 'Validation Rule Edit' screen for the 'Billing details and feedback' object. The 'Rule Name' is set to 'rating_should_be_less_than_5'. The 'Active' checkbox is checked. The 'Error Condition Formula' field contains the formula 'NOT(REGEX(Rating_for_service_c , "[1-5]{1}"))'. A tooltip for the 'NOT' function is visible, explaining it returns true if the formula expression is false. The 'Save' button is at the top right.

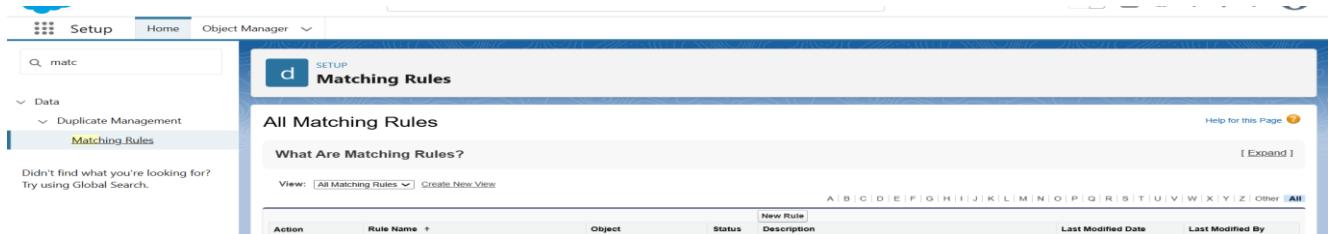
5. Enter the Error Message as “rating should be from 1 to 5”, select the Error location as Field and select the field as “Rating for Service”, and click Save.



MILESTONE 7: Duplicate Rule

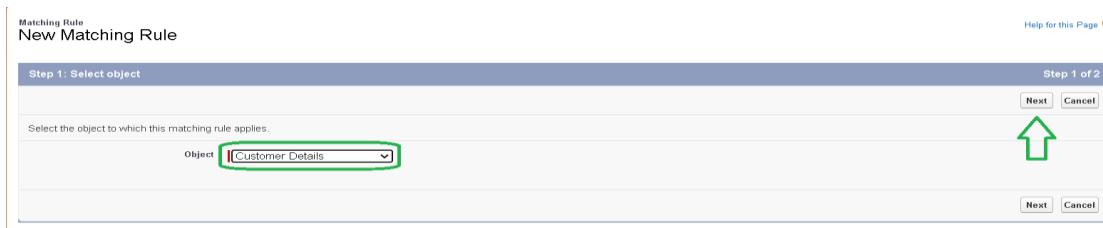
ACTIVITY 1: To create a matching rule to an Customer details Object

1. Go to quick find box in setup and search for matching Rule.
2. Click on matching rule >> click on New Rule.



The screenshot shows the 'Matching Rules' page in the Salesforce Setup. The page title is 'Matching Rules'. It displays a list of 'All Matching Rules'. The 'Object' column header is highlighted with a green arrow. The page includes a navigation bar at the top with links like 'Setup', 'Home', 'Object Manager', and a search bar. A sidebar on the left shows 'Data' and 'Duplicate Management' sections.

3. Select the object as Customer details and click Next.



The screenshot shows the 'Step 1: Select object' dialog. It has a title 'Matching Rule New Matching Rule'. Below it is a section titled 'Step 1 of 2'. A dropdown menu labeled 'Object' is open, showing 'Customer Details' which is highlighted with a green arrow. There are 'Next' and 'Cancel' buttons at the bottom right.

4. Give the Rule name : Matching customer details
5. Unique name : is auto populated
6. Define the matching criteria as
7. Field Matching Method
1. Gmail Exact
2. Phone Number Exact
8. Click save.
9. After Saving Click on Activate.



The screenshot shows the 'Rule Details' dialog. It has a 'Save' and 'Cancel' button at the top. The 'Object' field is set to 'Customer Details'. The 'Rule Name' field contains 'Matching Customer data'. The 'Unique Name' field contains 'matching_Customer_det'. In the 'Matching Criteria' section, there are two entries: 'Gmail' and 'Phone Number', both set to 'Exact' with 'AND' logic. At the bottom, there is an 'Add Filter Logic...' button and a 'Save' button with a green arrow pointing to it.

ACTIVITY 2: To create a duplicate rule to an Customer details Object

1. Go to quick find box in setup and search for Duplicate rules.
2. Click on Duplicate rule >> click on New Rule >> select customer details object.

The screenshot shows the 'Duplicate Rules' page in the Salesforce setup. The left sidebar has a 'Data' section with 'Duplicate Management' expanded, showing 'Duplicate Rules' selected. Below it are 'Duplicate Error Logs' and 'Matching Rules'. A global search bar at the top left says 'Q. Duplic...'. The main area is titled 'All Duplicate Rules' and contains a table of existing rules:

Rule Name	Description	New Rule	Matching Rule	Active	Last Modified By	Last Modified Date
Customer Detail duplicate	Identify accounts that duplicate other accounts	Customer Details	Matching Customer details	<input type="checkbox"/>	t2	10/10/2023
Standard Account Duplicate Rule	Identify contacts that duplicate other contacts and leads	Contact	Standard Contact Matching Rule	<input checked="" type="checkbox"/>	t2	24/09/2023
Standard Contact Duplicate Rule	Identify leads that duplicate other leads and contacts	Lead	Standard Lead Matching Rule	<input checked="" type="checkbox"/>	t2	24/09/2023
Standard Lead Duplicate Rule		Environment	Standard Contact Matching Rule	<input checked="" type="checkbox"/>	t2	24/09/2023
		Individual				
		Laptop				
		Lead				

3. Give the Rule name as : Customer Detail duplicate
4. Scroll a little in Matching rule section
5. Select the matching rule : Matching customer details
6. And Click on save.
7. After saving the Duplicate Rule, Click on Activate.

The screenshot shows the 'Edit Duplicate Rule' page for 'Customer Detail duplicate'. The page is divided into several sections:

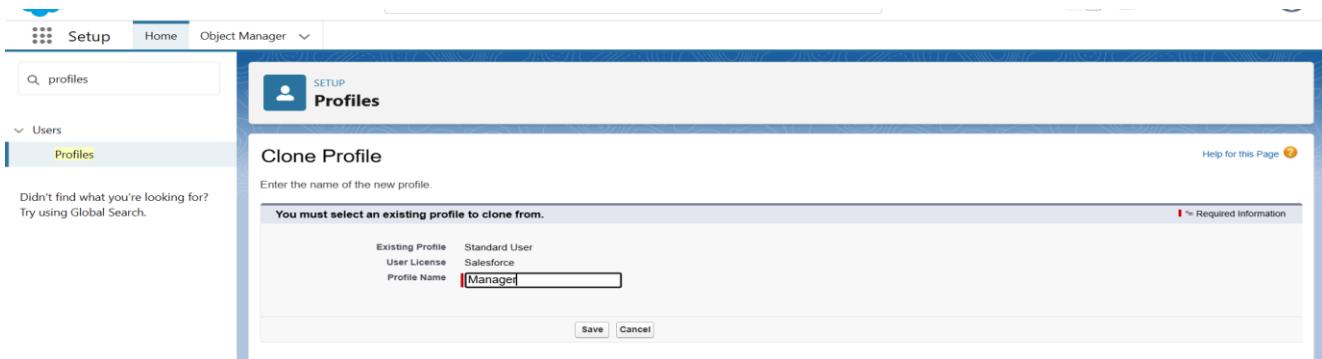
- Rule Details:** Shows the rule name 'Customer Detail duplicate' and a description field. It includes 'Object' (Customer Details) and 'Record Level Security' (Enforce sharing rules selected).
- Actions:** Allows configuration of actions on create and edit. For 'Action On Create', 'Allow' is selected with 'Alert' checked and 'Report' unchecked. For 'Action On Edit', 'Allow' is selected with 'Alert' unchecked and 'Report' checked. An 'Alert Text' field contains the message 'Use one of these records?'.
- Matching Rules:** Configures how records are identified. It shows 'Compare Customer Details With' set to 'Customer Details', 'Matching Rule' set to 'Matching Customer details', and 'Matching Criteria' set to '(Customer_Details__Email EXACT MatchBlank = FALSE) AND (Customer_Details__Phone_Number EXACT MatchBlank = FALSE) AND Mapping.Selected'. There are 'Add Rule' and 'Remove Rule' buttons.
- Conditions:** Allows specifying conditions a record must meet. It lists several fields with operators like '==None==' and values, separated by AND operators.

MILESTONE 8: Profiles

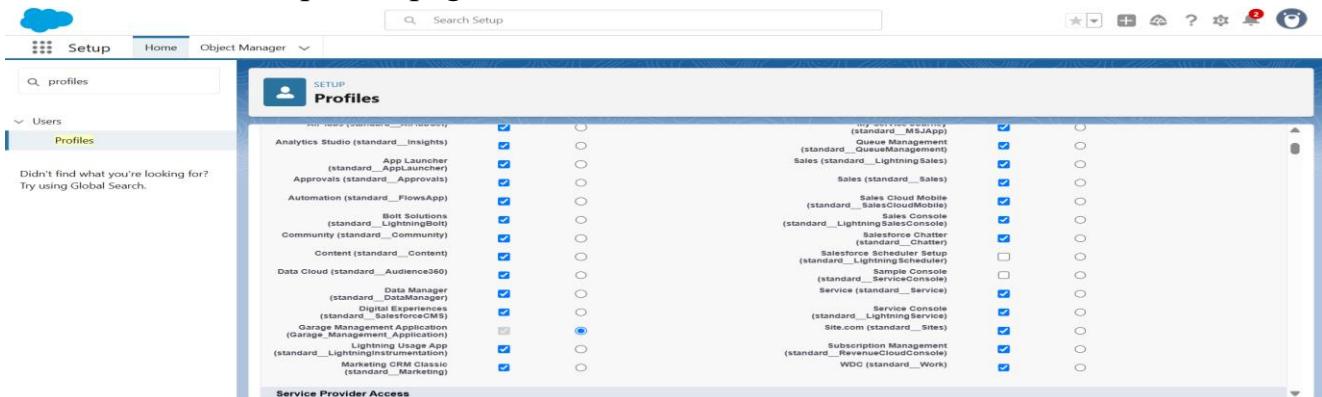
ACTIVITY 1: Manager Profile

1. To create a new profile:

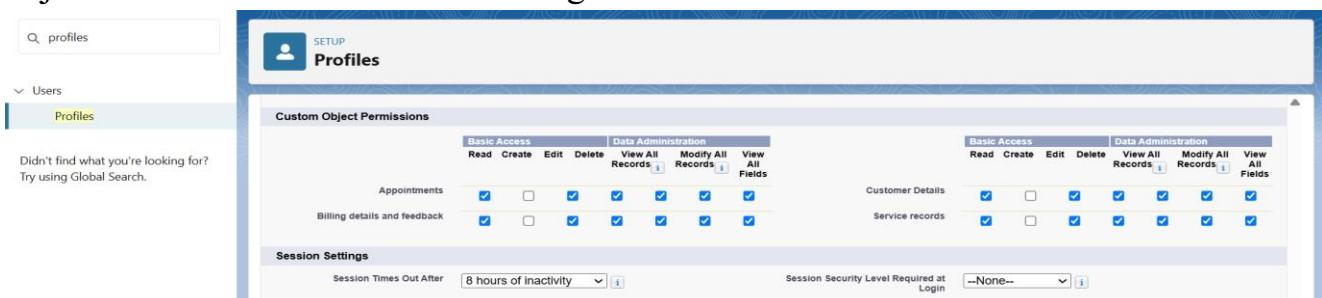
1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Standard User) >> enter profile name (Manager) >> Save.



2. While still on the profile page, then click Edit.



3. Select the Custom App settings as default for the Garage management.
4. Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.

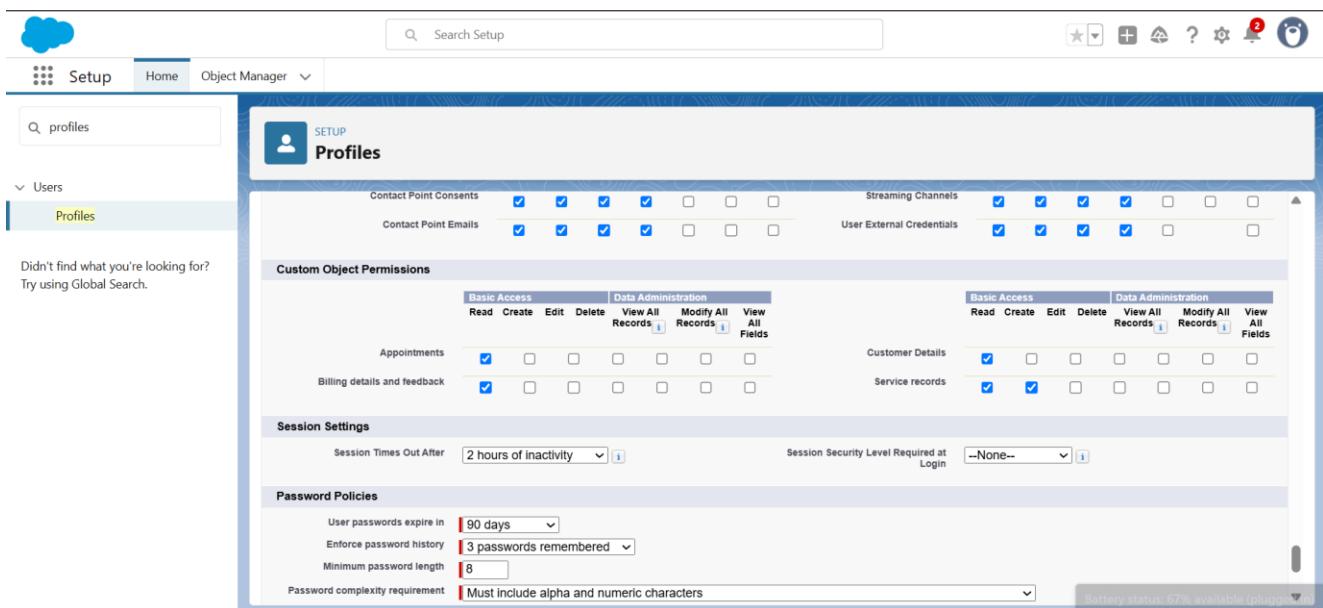


5. Changing the session times out after should be “ 8 hours of inactivity”.

6. Change the password policies as mentioned :
7. User passwords expire in should be “ never expires ”.
8. Minimum password length should be “ 8 ”, and click save.

ACTIVITY 1: Sales person Profile

1. Go to setup >> type profiles in quick find box >> click on profiles >> clone the desired profile (Salesforce Platform User) >> enter profile name (sales person) >> Save.
2. While still on the profile page, then click Edit.
3. Select the Custom App settings as default for the GArage management.
4. Scroll down to Custom Object Permissions and Give access permissions for Appointments,Billing details and feedback , service records and customer details objects as mentioned in the below diagram.



5. And click save.

MILESTONE 9: Role & Role Hierarchy

ACTIVITY 1: Creating manager role

Creating Manager Role:

1. Go to quick find >> Search for Roles >> click on set up roles.

<https://orgfarm-538bf0b53b-dev-ed.develop.lightning.force.com/lightning/setup/Roles/home>

2. Click on Expand All and click on add role under whom this role works.

javascript.roleTreeNode.expandAllRoles();

3. Give Label as “Manager” and Role name gets auto populated. Then click on Save.

ACTIVITY 2: Creating another role

Creating another two roles under manager

1. Go to quick find >> Search for Roles >> click on set up roles.
2. Click plus on CEO role, and click add role under manager.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes 'Setup', 'Home', and 'Object Manager'. A search bar at the top left contains the text 'roles'. On the left, a sidebar menu is open under 'Users', with 'Roles' selected. The main content area is titled 'SETUP Roles'. It displays a hierarchical list of roles:

- Add Role
 - CFO Edit | Del | Assign
 - COO Edit | Del | Assign
 - Manager Edit | Del | Assign
 - sales_person Edit | Del | Assign
 - SVP, Customer Service & Support Edit | Del | Assign
 - Customer Support, International Edit | Del | Assign
 - Customer Support, North America Edit | Del | Assign
 - Installation & Repair Services Edit | Del | Assign

3. Give Label as “sales person” and Role name gets auto populated. Then click on Save.

MILESTONE 10: Users

ACTIVITY 1: Create a user

1. Go to setup >> type users in quick find box >> select users >> click New user.
2. Fill in the fields
 - 1. First Name : Niklaus
 - 2. Last Name : Mikaelson
 - 3. Alias : Give a Alias Name
 - 4. Email id : Give your Personal Email id
 - 5. Username : Username should be in this form: text@text.text
 - 6. Nick Name : Give a Nickname
 - 7. Role : Manager

8. User licence : Salesforce
9. Profiles : Manager
10. Click on save

The screenshot shows the Salesforce Setup interface for creating a new user. The 'Users' section is selected in the sidebar. The main form is titled 'New User' and contains a 'User Edit' section with 'General Information'. The 'Role' field is set to 'Manager'. Other fields like 'User License' (Salesforce), 'Profile' (Manager), and 'Active' (checked) are also visible. The left sidebar shows the 'Users' section selected.

ACTIVITY 2: Create another user

1. Repeat the steps and create another user using
 - a. Role : sales person
 - b. User licence : Salesforce Platform
 - c. Profile : sales person

Note : create atleast 3 users with these permissions.

MILESTONE 11: Public Groups

ACTIVITY 1:Create a new public group

1. Go to setup >> type users in quick find box >> select public groups >> click New.

Setup Home Object Manager

Q public

SETUP
Public Groups

Public Groups Help for this Page

A public group is a set of users. It can contain individual users, other groups, the users in a particular role or territory, or the users in a role or territory plus all of the users below that role or territory in the hierarchy.

View: All | Edit | Create New View

Action	Label	Group Name	Created By	Created Date
Edit Del	sales team	sales_team	TS_Karthiyayini	8/30/2025, 6:48 AM

New

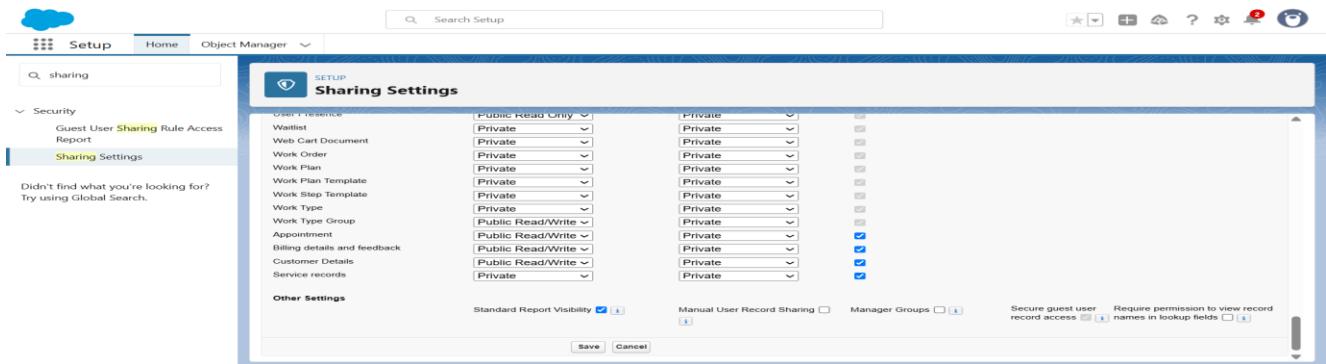
Didn't find what you're looking for?

2. Give the Label as “sales team”.
3. Group name is autopopulated.
4. Search for Roles.
5. In Available Members select Sales person and click on add it will be moved to selected member.
6. Click on save.

MILESTONE 12: Sharing Settings

ACTIVITY 1: Creating a sharing setting

1. Go to setup >> type users in quick find box >> select Sharing Settings >> click Edit.
2. Change the OWD setting of the Service records Object to private as shown in fig.



3. Click on save and refresh.
4. Scroll down a bit, Click new on Service records sharing Rules.
5. Give the Label name as “ Sharing setting”
6. Rule name is auto populated.
7. In step 3 : Select which records to be shared, members of “ Roles ” >> “ Sales person”
8. In step 4: share with, select “ Roles ” >> “ Manager ”
9. In step 5 : Change the access level to “ Read / write ”.
- 10.Click on save.



MILESTONE 13: Flows

ACTIVITY 1: Create a flow

1. Go to setup >> type Flow in quick find box >> Click on the Flow and Select the New Flow.

The screenshot shows the Salesforce Setup interface with the 'Flows' tab selected. A search bar at the top has 'flows' typed into it. On the left, a sidebar lists 'Process Automation' (Flows), 'Identity' (Login Flows), and a global search section. The main area is titled 'SETUP Flows' and shows a table titled 'Flow Definitions' with a dropdown 'All Flows'. The table has columns for 'Flow Label', 'Process Type', 'Actions', 'Triggers', 'Package State', 'Last Modified', and 'Last Modified By'. Several flows are listed, including 'Add or Modify Service Appoint...', 'Approvals Workflow: Evaluate...', 'Approvals Workflow: Process ...', 'Authentication Provider User ...', 'Basic Approval Request', 'BillingAmountFlow', 'Book Appointment from Invit...', and 'Cancel Item Flow'.

2. Select the Record-triggered flow and Click on Create.

The screenshot shows the 'Flow Builder' interface with the title 'New Automation'. On the left, a sidebar shows 'View All Automations' with 68 results, filtered by 'Triggered', 'Scheduled', 'Screen', and 'AutoLaunched'. A search bar at the top right contains 'record-triggered flow'. In the center, there's a section titled 'Types (19)' with three options: 'Screen Flow', 'Record-Triggered Flow' (which is highlighted with a blue border), and 'Schedule-Triggered Flow'. Below this is a 'Back' button. At the bottom, there are links for 'Get more on the AppExchange' and 'Dropbox promotion'.

3. Select the Object as “Billing details and feedback”in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimize the flow for: “Actions and Related Records” and Click on Done.
6. Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.
7. Give the Label Name : Amount Update
8. Api name : is auto populated
9. Set a filter condition : All Conditions are met(AND)
10. Field : Payment_Status__c

- 11.Operator : Equals
- 12.Value : Completed
- 13.And Set Field Values for the Billing details and feedback Record
- 14.Field : Payment_Paid_c
- 15.Value : {!\$Record.Service_records__r.Appointment__r.Service_Amount__c}
- 16.Click On Done.
- 17.Before creating another Element. Create a New Resource form Toolbox form top left.
- 18.Click on the New Resource, And select Variable.
- 19.Select the resource type as text template.
- 20.Enter the API name as “ alert”.
- 21.Change the view as Rich Text ? View to Plain Text.
- 22.In body field paste the syntax that given below.

Dear {!\$Record.Service_records__r.Appointment__r.Customer_Name__r.Name},

I hope this message finds you well. I wanted to take a moment to express my sincere gratitude for your recent payment for the services provided by our garage management team. Your prompt payment is greatly appreciated, and it helps us continue to provide top-notch services to you and all our valued customers.

Amount paid : {!\$Record.Payment_Paid_c}

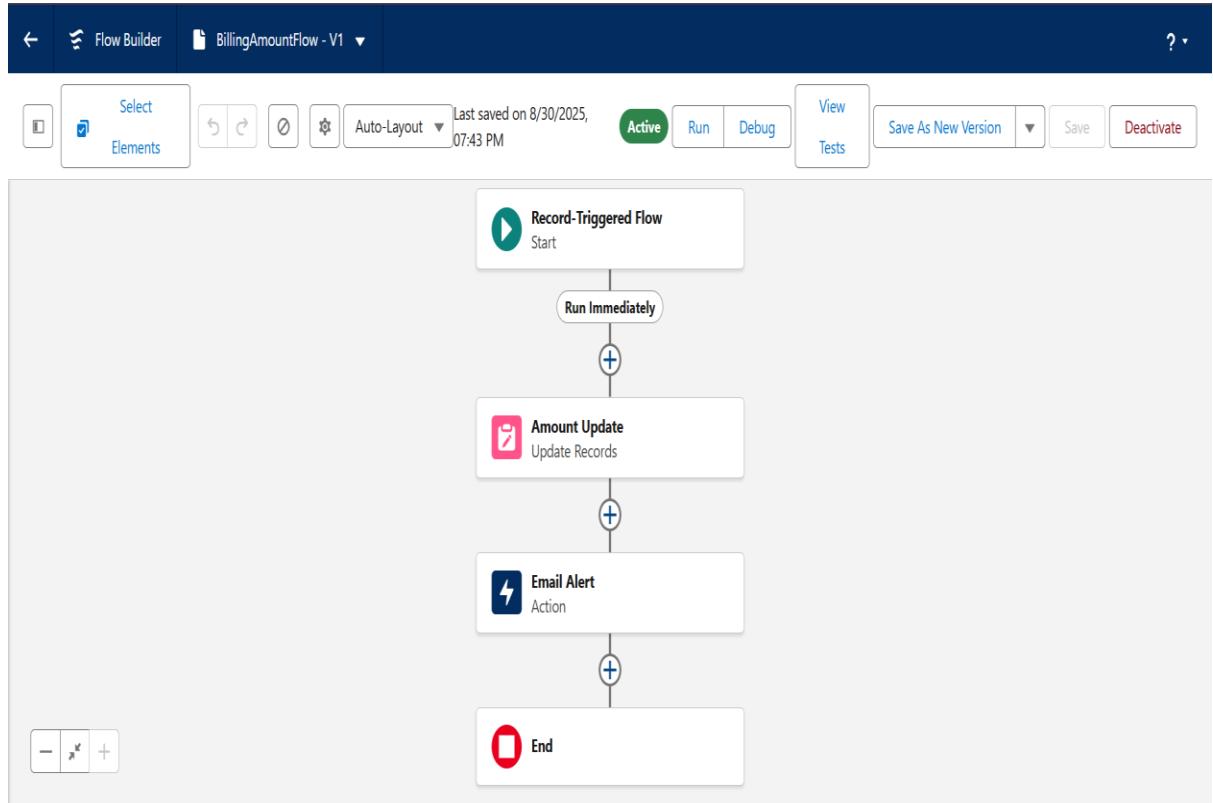
Thank you for Coming .

- 23.Click done.

- 24.Now Click on Add Element , select Action.
- 25.Their action bar will be opened in that search for “ send email ” and click on it.
- 26.Give the label name as “ Email Alert”
- 27.API name will be auto populated.
- 28.Enable the body in set input values for the selected action.
- 29.Select the text template that created , Body : {!alert}
- 30.Include recipient address list select the email form the record.
- 31.RecipientAddressList:
- {!\$Record.Service_records__r.Appointment__r.Customer_Name__r.Gmail_c}
- 32.Include subject as “ Thank You for Your Payment - Garage Management”.
- 33.Click done.

34.Click on save. Give the Flow label , Flow Api name will be autopopulated.

35.And click save, and click on activate.

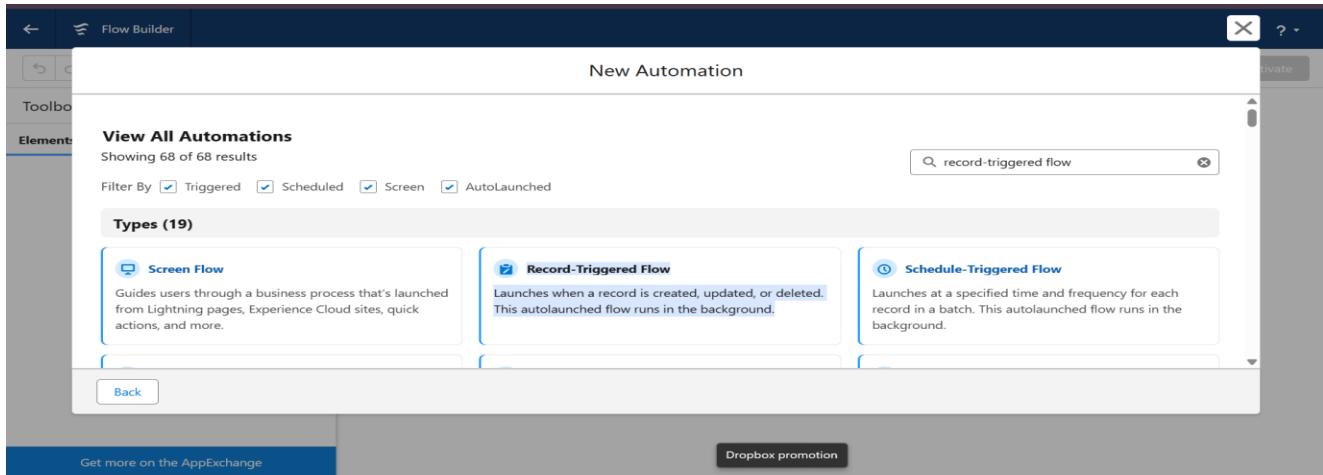


ACTIVITY 2: Create another flow

1. Go to setup ? type Flow in quick find box ? Click on the Flow and Select the New Flow.

The screenshot shows the Salesforce Setup interface. The top navigation bar includes a cloud icon, 'Setup', 'Home', and 'Object Manager'. A search bar says 'Search Setup' and a toolbar with various icons. On the left, a sidebar has a search bar 'Q. flows' and sections for 'Process Automation' (with 'Flows' selected) and 'Identity' (with 'Login Flows'). A message says 'Didn't find what you're looking for? Try using Global Search.' The main content area is titled 'Flows' and shows a table of 'Flow Definitions'. The table has columns: 'Flow Label ↑', 'Process Type', 'A...', 'Te...', 'Package State', 'Pa...', 'Last Mo...', 'Last Modifie...', and 'Actions'. There are 50+ items, sorted by flow label, filtered by all flow definitions, and updated a few seconds ago. One row is highlighted: 'BillingAmountFlow' (Autolaunched Flow, Unmanaged, Karthiyayini ... 8/30/2025, 7:13 A...). Other rows include 'Add or Modify Service Appoi...', 'Approvals Workflow: Evaluate...', 'Approvals Workflow: Process ...', 'Authentication Provider User ...', 'Basic Approval Request', 'Flow Orchestration for CMS', 'Book Appointment from Invit...', 'Cancel Item Flow', and 'Salesforce Scheduler Flow'.

2. Select the Record-triggered flow and Click on Create.



3. Select the Object as “Service records” in the Drop down list.
4. Select the Trigger Flow when: “A record is Created or Updated”.
5. Select the Optimise the flow for: “Actions and Related Records” and Click on Done.
6. Under the Record-triggered Flow Click on “+” Symbol and In the Drop down List select the “Update records Element”.
7. Set a filter condition : All Conditions are met(AND)
8. Field : **Quality_Check_Status_c**
9. Operator : **Equals**
10. Value : **True**
11. And Set Field Values for the Billing details and feedback Record
12. Field : **Service_Status_c**
13. Value : **Completed**
14. Click On **Done**.
15. Click on **save**
16. Given the Flow label as **Update Service Status** , Flow Api name will be auto populated.
17. And click save, and click on **activate**.

Set Filter Conditions

Condition Requirements to Update Record

All Conditions Are Met (AND)

Field	Operator	Value
Quality_Check_Status__c	Equals	<input type="checkbox"/> True <input type="button" value="X"/>

[+ Add Condition](#)

Set Field Values for the Service record Record

Field

Service_Status__c

Value

Completed



[+ Add Field](#)

MILESTONE 14:Apex Trigger

ACTIVITY 1: Apex Handler

UseCase : This use case works for Amount Distribution for each Service the customer selected for there Vehicle.

1. Login to the respective trailhead account and navigate to the gear icon in the top right corner.
2. Click on the Developer console. Now you will see a new console window.
3. In the toolbar, you can see FILE. Click on it and navigate to new and create New apex class.
4. Name the class as “AmountDistributionHandler”.

The screenshot shows the Salesforce Setup interface. In the center, the 'Flows' section displays a list of flow definitions. On the right, a context menu is open over a specific row in the table, listing options like 'Open link in new tab', 'Save link as...', and 'Inspect'. Below the flows, a code editor window titled 'AmountDistributionHandler.apc' shows Java code for handling appointment amounts based on service types.

Flow Label	Process Type	Package State	Last Modified By
Add or Modify Service Appoi...	Salesforce Scheduler Flow	Managed-Installed	
Approvals Workflow: Evaluate...	Screen Flow	Managed-Installed	
Approvals Workflow: Process ...	Screen Flow	Managed-Installed	
Authentication Provider User ...	Identity User Registration Fl...	Managed-Installed	
Basic Approval Request	Flow Orchestration for CMS	Managed-Installed	
BillingAmountFlow	Autolaunched Flow	Unmanaged	Karthiyayini ...
Book Appointment from Invit...	Salesforce Scheduler Flow	Managed-Installed	
Cancel Item Flow	Screen Flow	Managed-Installed	

```

public class AmountDistributionHandler {
    public static void amountDist(List<Appointment__c> listApp){
        List<Service_records__c> serList = new List <Service_records__c>();
        for(Appointment__c app : listApp){
            if(app.Maintenance_service__c == true && app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 10000;
            }
            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
                app.Service_Amount__c = 5000;
            }
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 8000;
            }
            else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
                app.Service_Amount__c = 7000;
            }
            else if(app.Maintenance_service__c == true){
                app.Service_Amount__c = 2000;
            }
            else if(app.Repairs__c == true){
                app.Service_Amount__c = 3000;
            }
            else if(app.Replacement_Parts__c == true){
                app.Service_Amount__c = 5000;
            }
        }
    }
}

```

Code:

```

public class AmountDistributionHandler {

    public static void amountDist(List<Appointment__c> listApp){
        List<Service_records__c> serList = new List <Service_records__c>();

        for(Appointment__c app : listApp){
            if(app.Maintenance_service__c == true && app.Repairs__c == true &&
app.Replacement_Parts__c == true){
                app.Service_Amount__c = 10000;
            }
            else if(app.Maintenance_service__c == true && app.Repairs__c == true){
                app.Service_Amount__c = 5000;
            }
        }
    }
}

```

```

    }
    else if(app.Maintenance_service__c == true && app.Replacement_Parts__c ==
true){
        app.Service_Amount__c = 8000;
    }
    else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
        app.Service_Amount__c = 7000;
    }
    else if(app.Maintenance_service__c == true){
        app.Service_Amount__c = 2000;
    }
    else if(app.Repairs__c == true){
        app.Service_Amount__c = 3000;
    }
    else if(app.Replacement_Parts__c == true){
        app.Service_Amount__c = 5000;
    }
}

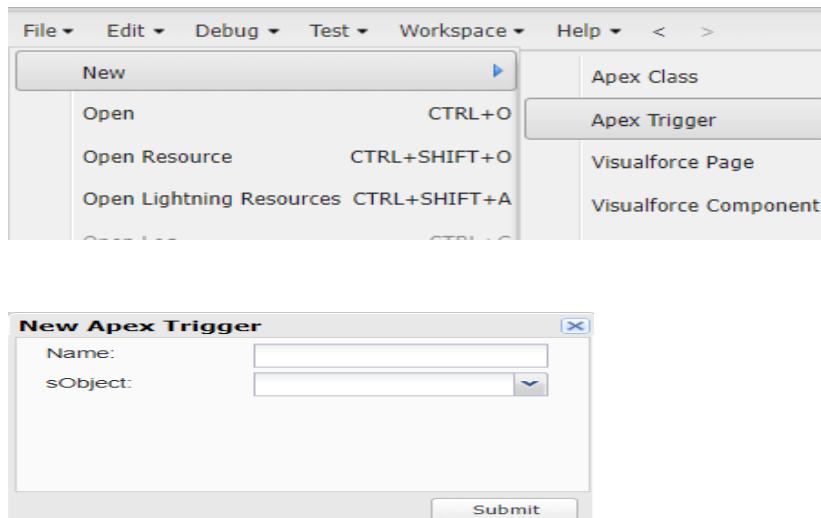
}
}
}
}

```

Trigger Handler :

How to create a new trigger :

1. While still in the trailhead account, navigate to the gear icon in the top right corner.
2. Click on developer console and you will be navigated to a new console window.
3. Click on File menu in the tool bar, and click on new? Trigger.
4. Enter the trigger name and the object to be triggered.
5. Name : AmountDistribution
6. sObject : Appointment__c



Syntax For creating trigger :

The syntax for creating trigger is :

Trigger [trigger name] on [object name](Before/After event)

```
{  
}
```

In this project , trigger is called whenever the particular records sum exceed the threshold i.e minimum business requirement value. Then the code in the trigger will get executed.

1. Handler for the Appointment Object

The screenshot shows the Salesforce IDE interface. The top menu bar includes File, Edit, Debug, Test, Workspace, Help, and various navigation icons. The tabs at the top show 'AmountDistributionHandler.apxc' and 'AmountDistribution.apxt', with 'AmountDistribution.apxt' being the active tab. Below the tabs, there's a dropdown for 'Code Coverage: None' and 'API Version: 64'. A 'Go To' button is also present. The main area displays the Apex trigger code:

```
1 trigger AmountDistribution on Appointment__c (before insert, before update) {
2
3     if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
4         AmountDistributionHandler.amountDist(trigger.new);
5
6     }
7 }
```

Code:

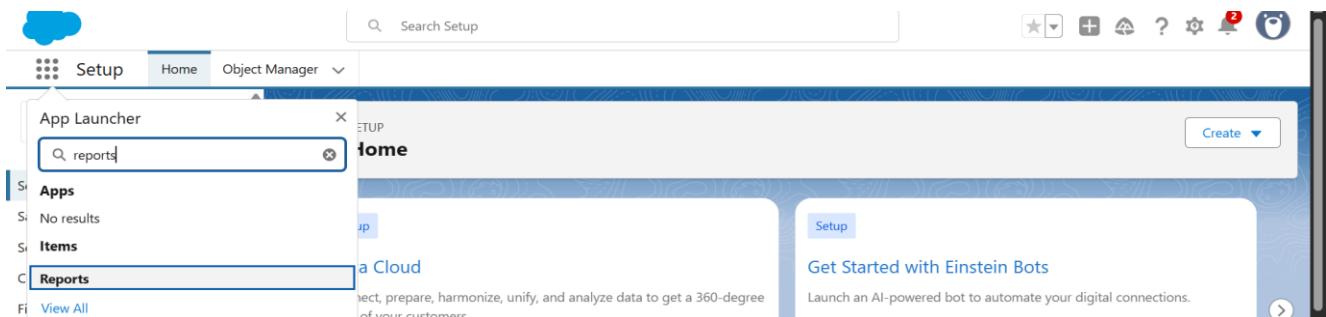
```
trigger AmountDistribution on Appointment__c (before insert, before update) {

    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);
    }
}
```

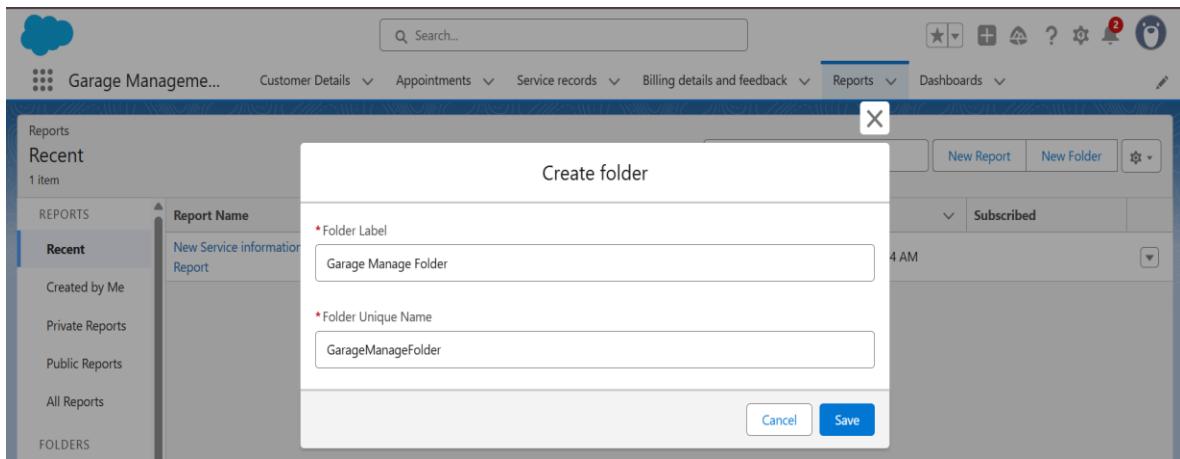
MILESTONE 15:Reports

ACTIVITY 1: Create a report folder

1. Click on the app launcher and search for reports.
2. Click on the report tab, click on new folder.

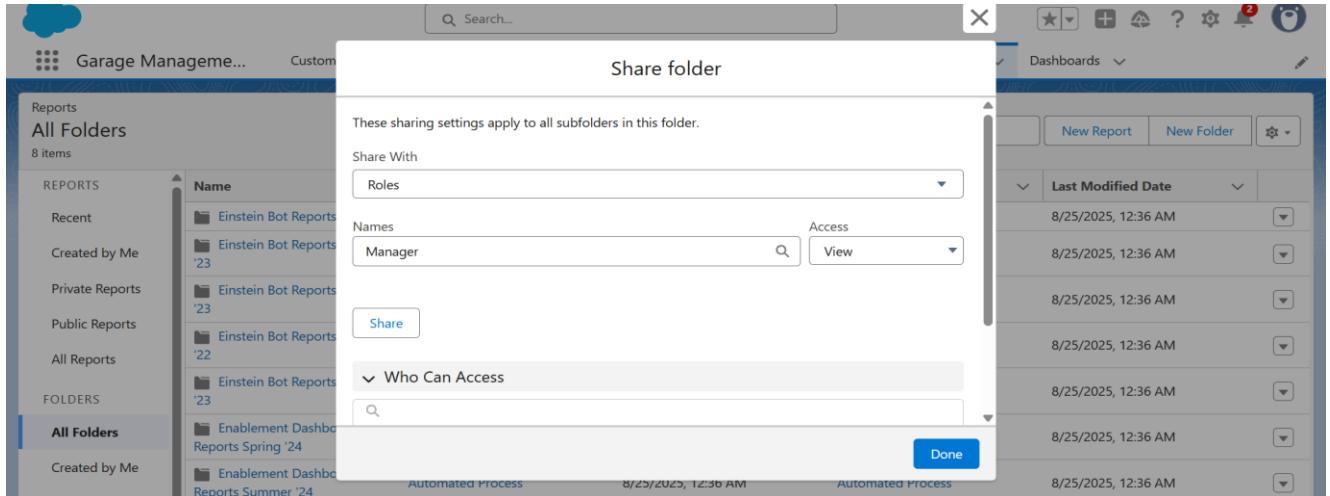


3. Give the Folder label as “Garage Management Folder”, Folder unique name will be auto populated.
4. Click save.



ACTIVITY 2: Sharing a report folder

1. Go to the app >> click on the reports tab.
2. Click on the All folder , click on the Drop down arrow for Garage Management folder, and Click on share.
3. Select the share with as “roles”, in name field search for “manager”, give “view” as access for that role.
4. Then click share, and click on Done.



ACTIVITY 3: Create a report type

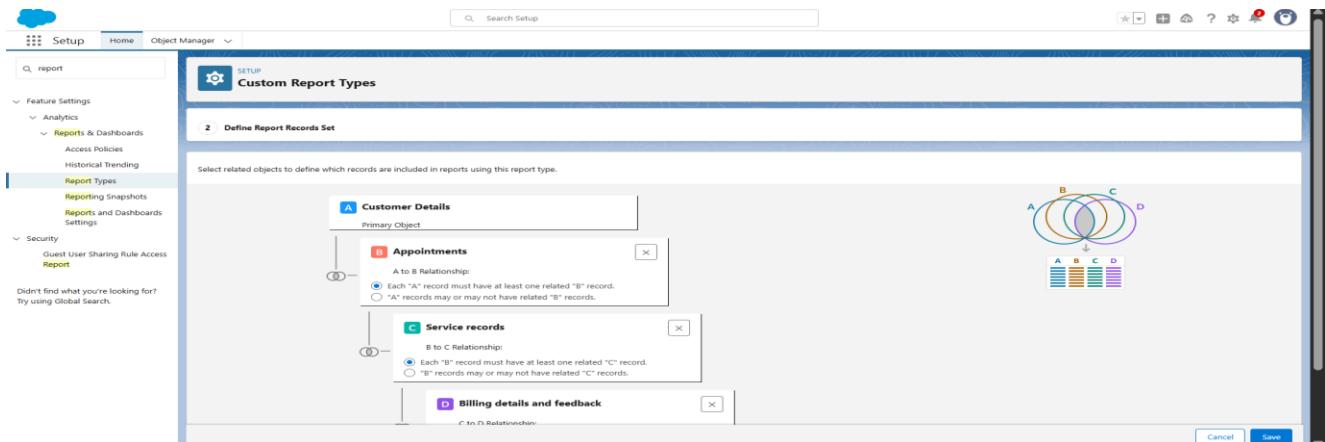
1. Go to setup >> type users in quick find box >> select Report Type >> click on Continue.
2. Click on new custom report type.

The screenshot shows the Salesforce Setup interface with the 'Report Types' page selected. The sidebar on the left includes links for Feature Settings, Analytics, Reports & Dashboards, and Report Types. A green arrow highlights the 'Report Types' link. The main content area displays a list of 'All Custom Report Types' with columns for Action, Label, Description, Category, Deployed, Created By Alias, and Created Date. A green arrow points to the 'New Custom Report Type' button at the top of the list.

3. Select the Primary object as “ Customer details” .
4. Give the Report type Label as “ Service information ”
5. Report type Name is autopopulated.
6. Keep the Description as same.
7. Select Store in Category as “ other Reports ”
8. Select the deployment status as “ Deployed ”, click on Next.

The screenshot shows the 'New Custom Report Type' wizard. The 'Report Type Focus' section has 'Customer Details' selected as the primary object. The 'Identification' section shows 'Service information' in the Report Type Label field, 'Service_information' in the Report Type Name field, and 'Service information' in the Description field. The 'Store in Category' dropdown is set to 'Other Reports'. In the 'Deployment' section, the 'Deployed' radio button is selected. A green arrow points to each of these highlighted fields. At the bottom right, there is a 'Next' button with a green arrow pointing to it.

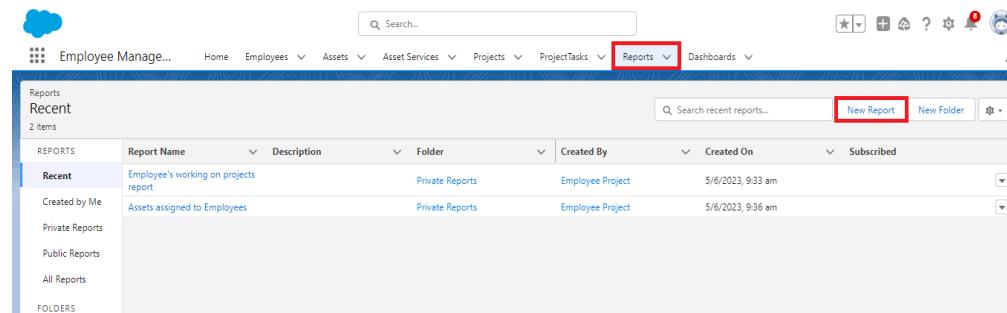
9. now , Click on Related object box.
- 10.Click on Select Object, choose Appointment Object as
- 11.Again Click to relate another object.
- 12.And select the related object as “ service records”.
- 13.Repeat the process and select the related object as “ Billing details and feedback”.
- 14.And click on save.



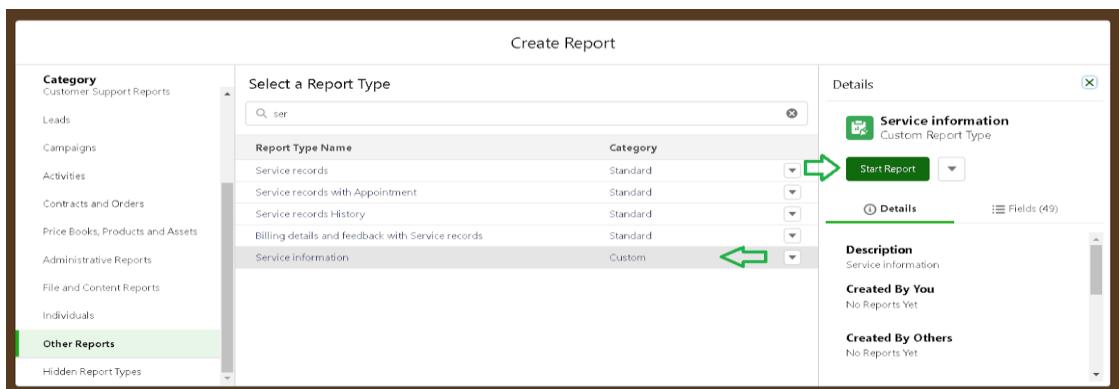
ACTIVITY 4: Create a report

Note : Before creating report, create latest “10” records in every object.
Try to fill every field in each record for better experience.

1. Go to the app >> click on the reports tab
2. Click New Report.



3. Select the Category as other reports, search for Service Information, select that report, click on it. And click on start report.



4. Their outline pane is opened already, select the fields that mentioned below in column section.
 - a. Customer name
 - b. Appointment Date
 - c. Service Status
 - d. Payment paid
5. Remove the unnecessary fields.
6. Select the fields that mentioned below in GROUP ROWS section.
 - a. Rating for Service
7. Select the fields that mentioned below in GROUP ROWS section.
 - a. Payment Status
8. Click on Add Chart , Select the Line Chart.
9. Click on save, Give the report Name : New Service information Report
10. Report unique Name is auto populated.
11. Select the folder the created and Click on save.

The screenshot shows the Salesforce Report Builder interface. At the top, there's a navigation bar with tabs like 'Customer Details', 'Appointments', 'Service records', 'Billing details and feedback', 'Reports' (which is selected), and 'Dashboards'. Below the navigation bar, there's a search bar and a toolbar with buttons for 'Save & Run', 'Save', 'Close', and 'Run'.

The main area is titled 'New Service information Report'. It displays a table titled 'Service Information' with the following data:

	Rating for service	Payment Status	Completed	Total
3	Sum of Payment Paid Record Count	\$15,000 2	\$15,000 2	\$15,000 2
4	Sum of Payment Paid Record Count	\$30,000 6	\$30,000 6	\$30,000 6
5	Sum of Payment Paid Record Count	\$10,000 2	\$10,000 2	\$10,000 2
Total	Sum of Payment Paid Record Count	\$55,000 10	\$55,000 10	\$55,000 10

Below the table is a line chart titled 'Rating for service' with data points at 3, 4, and 5. The Y-axis ranges from 0 to 40k. The chart shows values approximately 15k at 3, 30k at 4, and 10k at 5.

On the left side, there's a sidebar with sections for 'Fields', 'Groups', 'Rating for service', 'Service information', and 'Payment Status'. Under 'Fields', there are buttons for 'Add group...' and 'Add column...'. Under 'Groups', there's a 'GROUP ROWS' button. Under 'Rating for service', there's a 'Rating for service' button. Under 'Service information', there's a 'Customer Name', 'Appointment Date', 'Service Status', and 'Payment Paid' button. Under 'Payment Status', there's a 'Payment Status' button.

At the bottom of the screen, there's a taskbar with icons for news, search, and various applications, along with system status indicators like battery level (70%), network connection, and date/time (06-09-2025).

The second screenshot shows the 'Save Report As' dialog box overlaid on the report builder interface. The dialog has a title 'Save Report As' and contains the following fields:

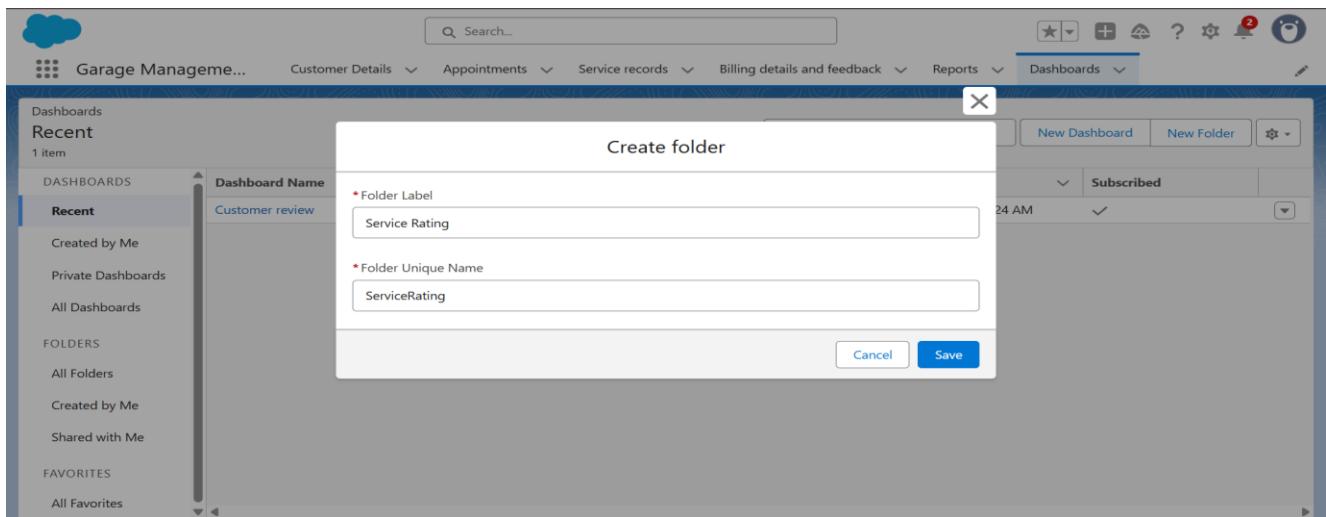
- *Report Name: New Service Information Report
- Report Unique Name: New_Service_information_Report_EzA
- Report Description: (empty)
- Folder: Garage Management Folder

At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

MILESTONE 16:Dashboards

ACTIVITY 1:Create a dashboard folder

1. Click on the app launcher and search for dashboard.
2. Click on dashboard tab.
3. Click new folder, give the folder label as “ Service Rating dashboard”.
4. Folder unique name will be auto populated.
5. Click save.



6. Follow the same steps, from Reports Milestone and Activity 2, and provide the sharing settings for the folder that was just created.

ACTIVITY 1:Create dashboard

1. Go to the app >> click on the Dashboards tabs.
2. Give a Name and select the folder that created, and click on create.

New Dashboard

* Name
 Customer review

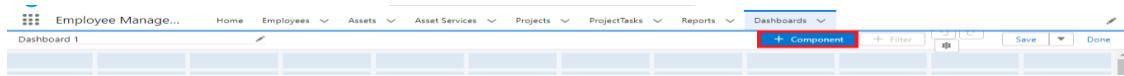
Description

Folder
 Select Folder

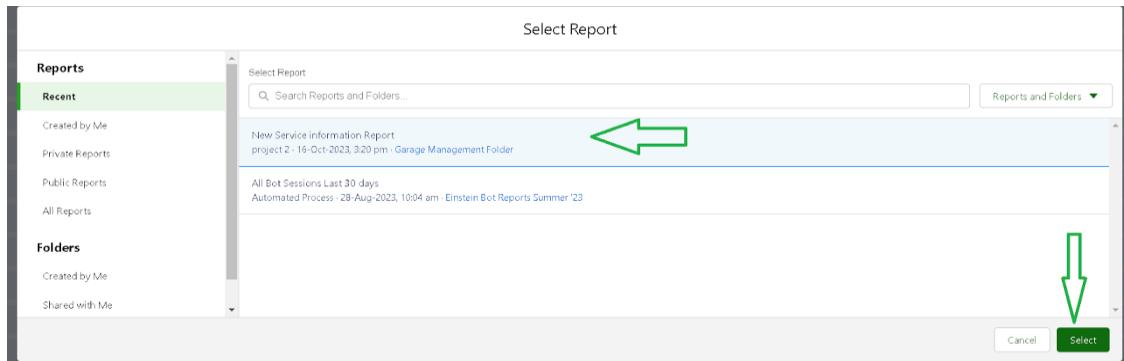
Cancel Create



3. Select add component.



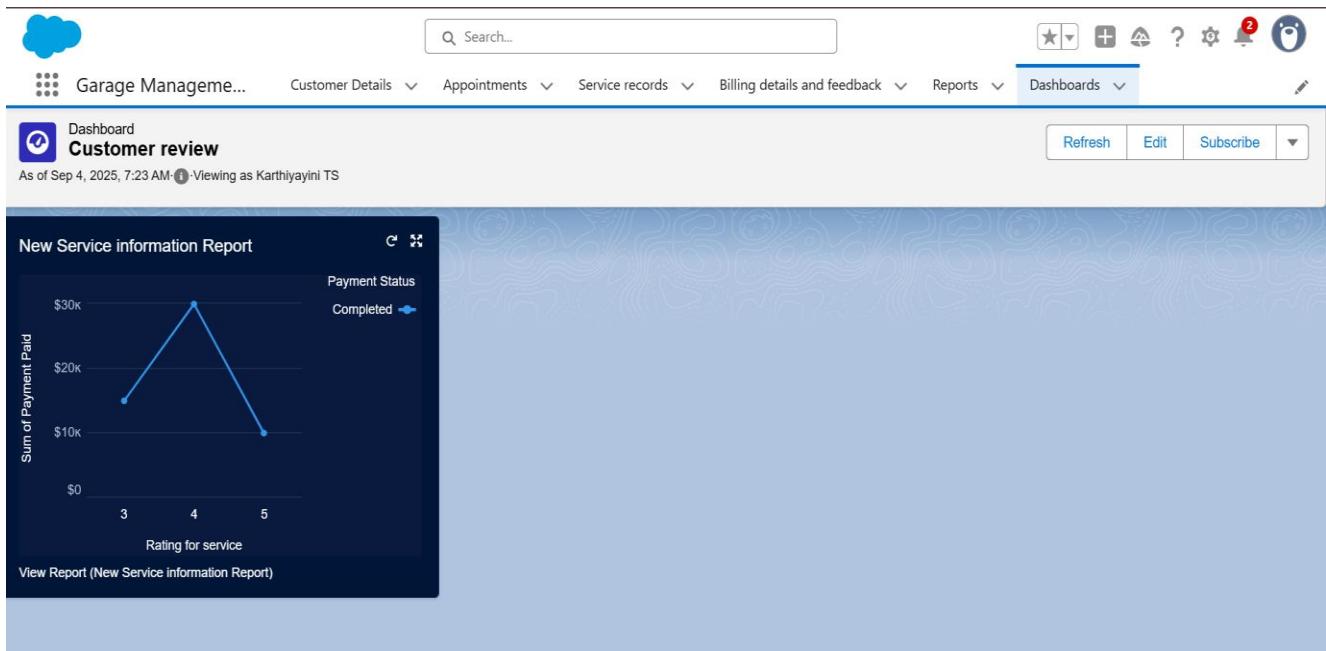
4. Select a Report and click on select.



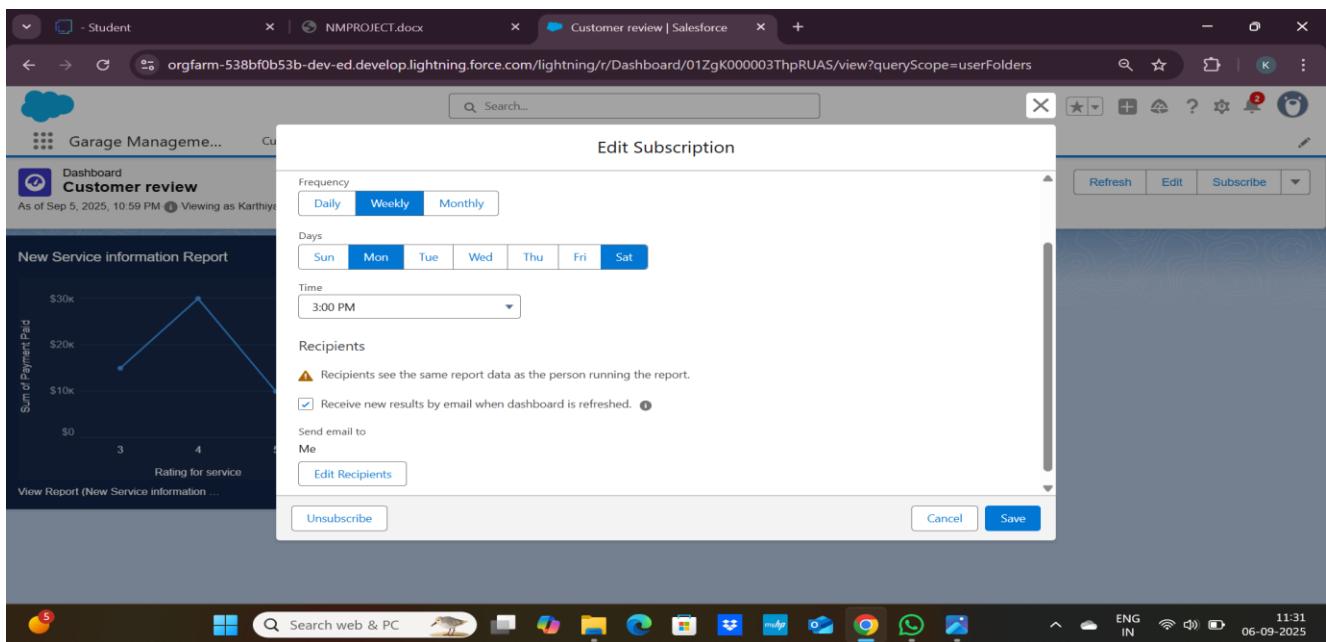
5. Select the Line Chart. Change the theme.

6. Click Add then click on Save and then click on Done.

7. Preview is shown below.



8. After that Click on Subscribe on top right.
9. Set the Frequency as “ weekly ”.
10. Set a day as monday.
11. And Click on save.



MILESTONE 17: User Adoption

ACTIVITY 1: Creating records

To create a record in the follow objects follow these steps

1. Click on the app launcher located at the left side of the screen.
2. Search for “**Garage Management**” and click on it.
3. Click on the “**Consumer details tab**”.
4. Click on new and fill the details as shown below figs, and click save.

New Customer Details

Information

*Customer Name: Mac

Phone number: 9876543210

Gmail: mac@gmail.com

Owner: Karthiyayini TS

Buttons: Cancel, Save & New, Save

Now, Create the Appointment Record

- 1.Click on the “**Appointment tab**”.
- 2.Enter the customer details as created, while entering Appointment Date enter the date less than the created date.
- 3.Match the validation while entering the vehicle number plate.
- 4.Select the services you need.
- 5.Click on save to see the Service Amount.

New Appointment

Information

Appointment Name: app-012

Appointment Date: 8/31/2025

Maintenance service:

Repairs:

Replacement Parts:

Service Amount: \$2,000

Vehicle number plate: TNO ITS3490

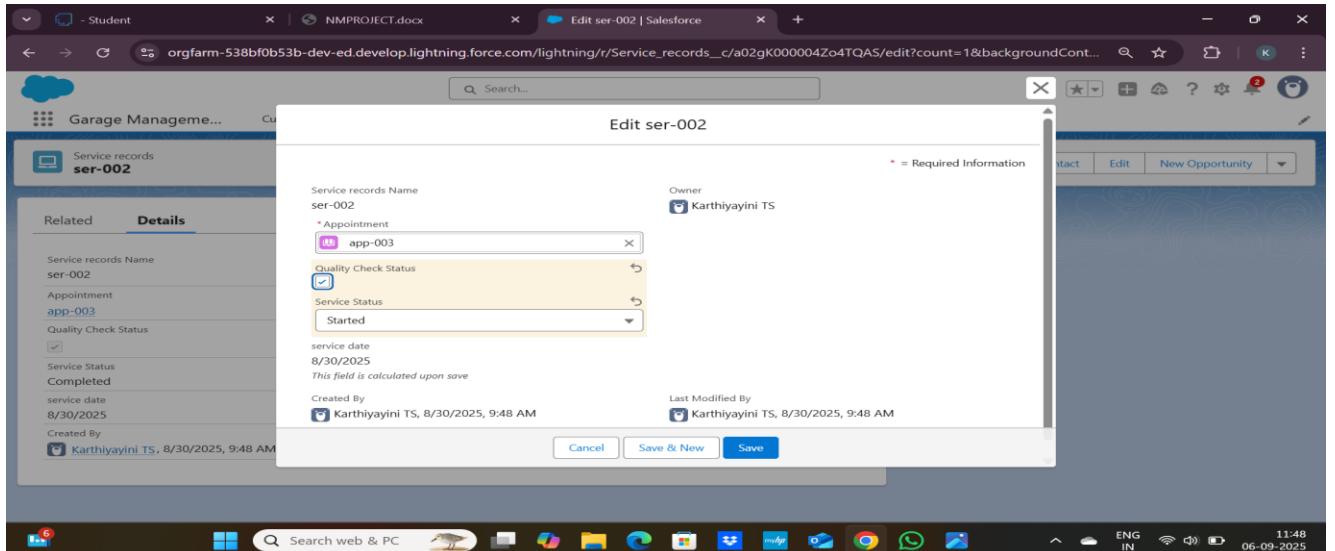
Owner: Karthiyayini TS

Buttons: Cancel, Save & New, Save

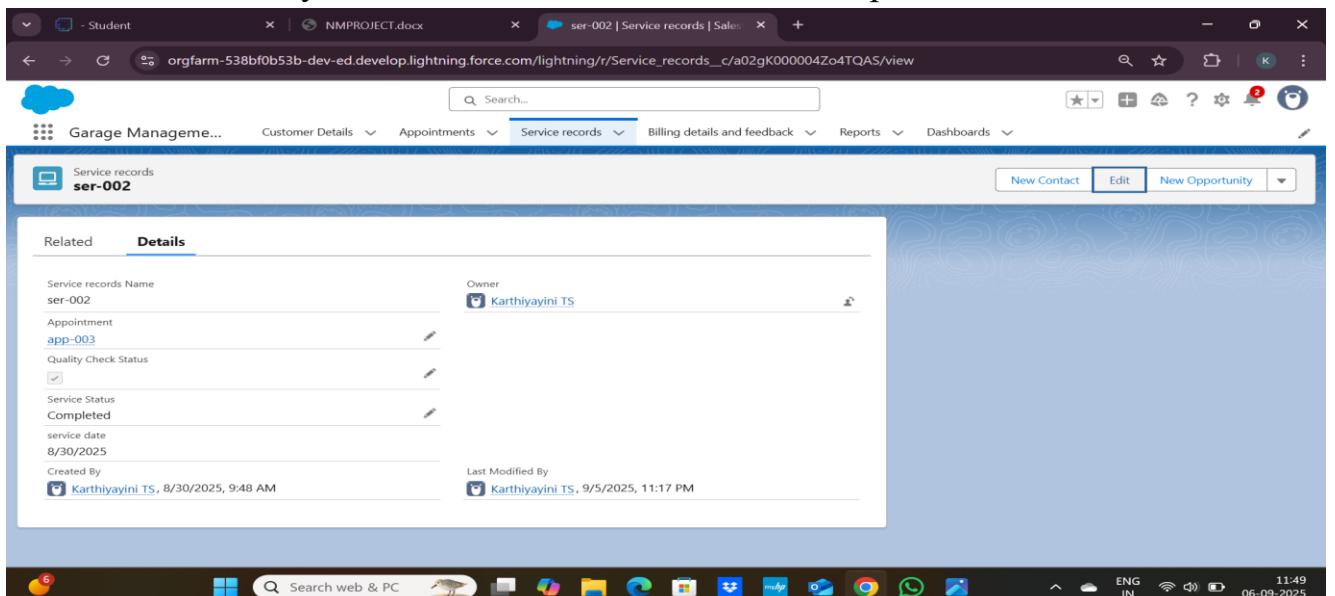
Now, Create a service Record

1. Click on the “**Service record tab**”.
2. Enter the Appointment, and started is selected as default.
3. Click on save.

4. Open the record and click on Quality check status as true.
5. Click on save.



6. Now automatically Service status will be moved to completed.



NOTE : Thus all the milestone are created successfully using the above instructions.

PHASE 3

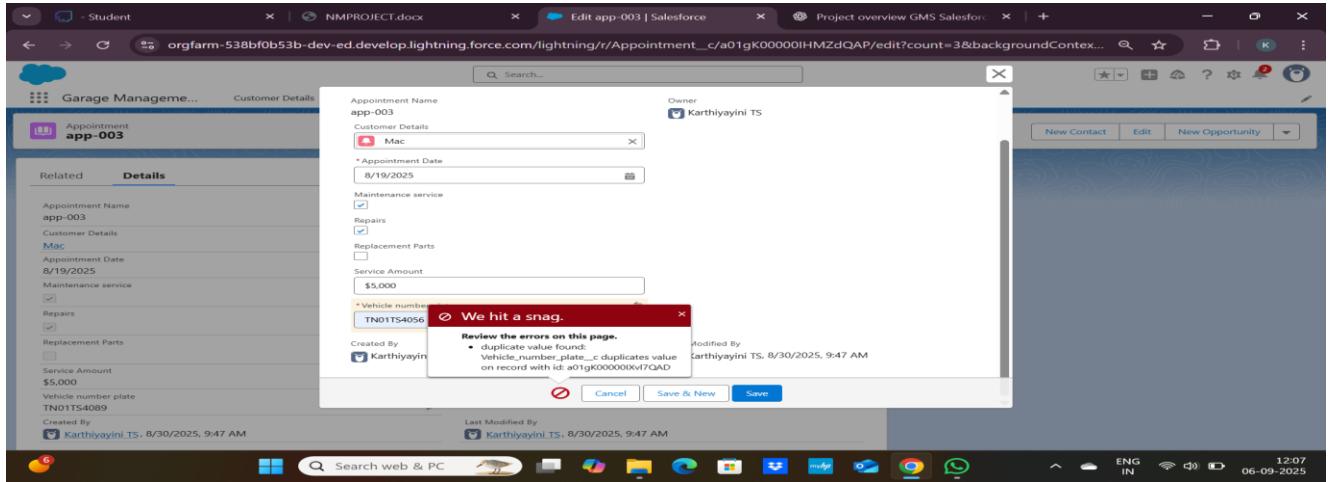
TESTING PHASE

FUNCTIONAL AND NON-FUNCTIONAL TESTING

Performance Testing

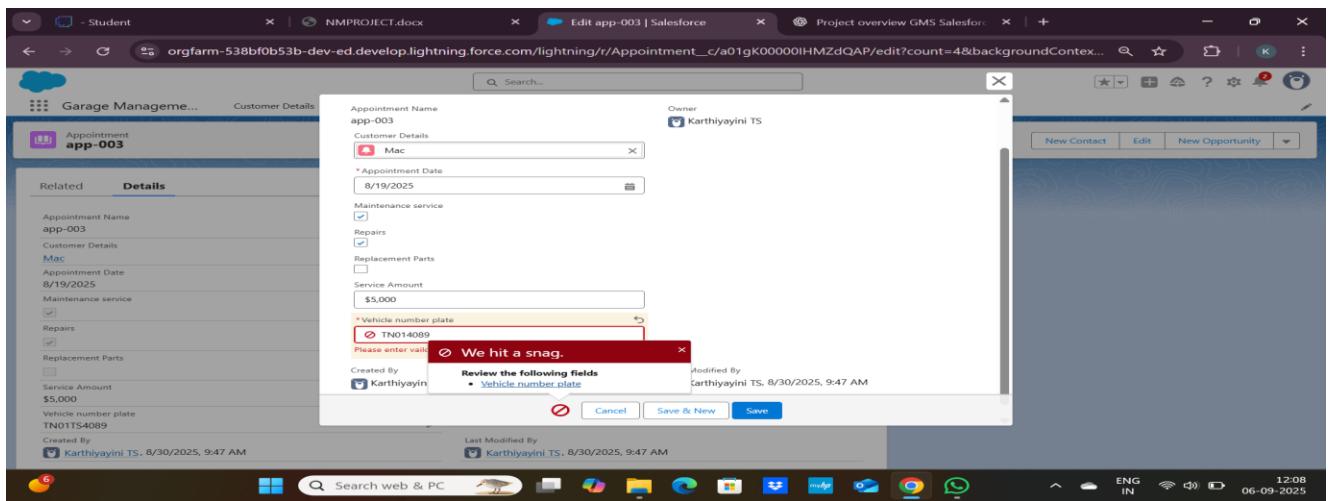
❖ Duplicate Vehicle Record:

Trigger validation by entering duplicate vehicle number records for the same customer.



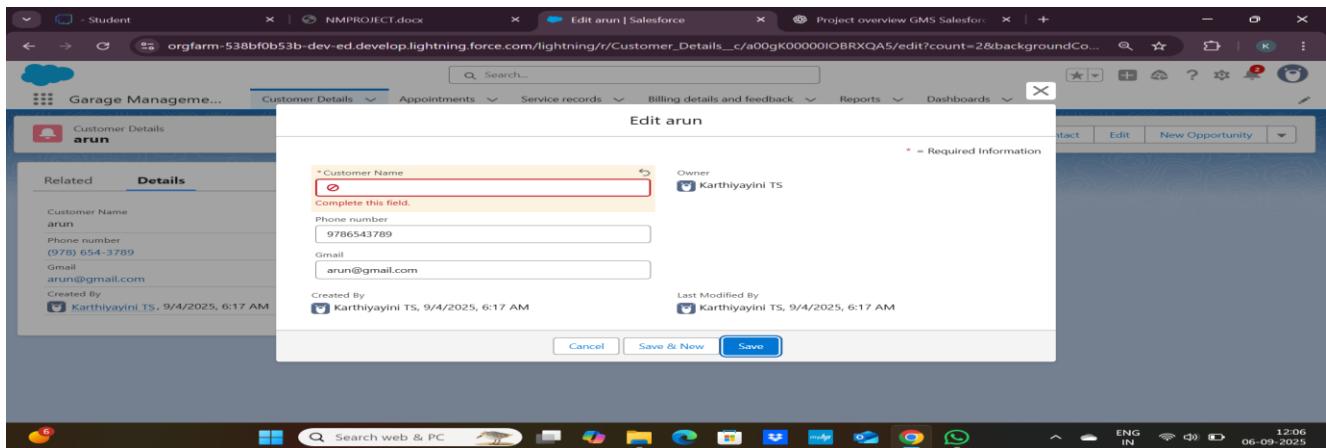
❖ Invalid Vehicle Number Format:

Trigger validation by entering a wrong or invalid number plate format.



❖ Empty Customer Details:

Trigger validation by leaving mandatory fields like customer name or contact number blank.



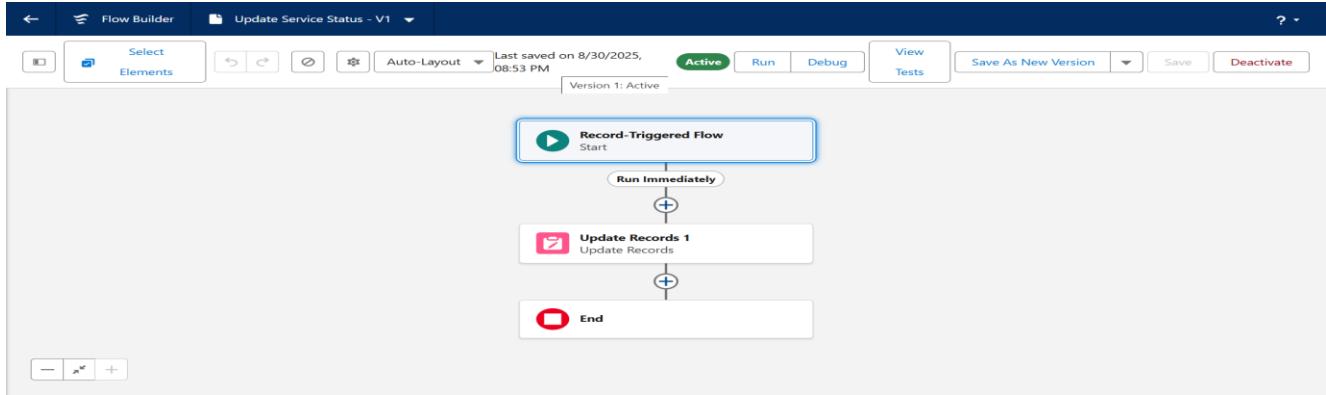
RESULT

OUTPUT SCREENSHOTS

- Tabs for Appointments,Billing details and feedback , customer details and service records.

Action	Label	Tab Style	Description
Edit Del	Appointments		Books
Edit Del	Billing_details_and_feedback		Computer
Edit Del	Customer_Details		Bell
Edit Del	Service_records		Laptop

- Flows runs



- Reports

Service information

Rating for service	Payment Status	Completed	Total
3	Sum of Payment Paid	\$15,000	\$15,000
	Record Count	2	2
4	Sum of Payment Paid	\$30,000	\$30,000
	Record Count	6	6
5	Sum of Payment Paid	\$10,000	\$10,000
	Record Count	2	2
Total	Sum of Payment Paid	\$55,000	\$55,000
	Record Count	10	10

Details (10 Rows)

Customer Name	Appointment Date	Service Status	Payment Paid	
Browne E	8/29/2025	Completed	\$10,000	
rahul	8/19/2025	Completed	\$5,000	
3	Jack	8/31/2025	Completed	\$5,000
4	Mac	8/19/2025	Completed	\$5,000
			Grand Total	\$30,000
			Stacked Summaries	

Line Chart: Sum of Payment Paid vs Rating for service

- Dashboards and the output line chart

New Service information Report

Payment Status: Completed

Sum of Payment Paid

Rating for service

Line Chart Data:

Rating for service	Sum of Payment Paid
3	\$15,000
4	\$30,000
5	\$10,000

CONCLUSION

The Garage Management System project demonstrates how Salesforce CRM can be applied to optimize garage operations. By automating tasks

such as service scheduling, inventory tracking, and billing, the solution improves both efficiency and customer experience. This project highlights the importance of cloud-based solutions in transforming traditional business operations into scalable, data-driven systems.

- **Mobile Application Integration:** Extend the system with a mobile app for customers to book services, track status, and receive notifications in real time.
- **Payment Gateway Integration:** Enable secure online payments through UPI, debit/credit cards, and digital wallets.
- **AI-Powered Predictive Maintenance:** Use AI/ML in Salesforce to predict vehicle service needs and provide proactive reminders to customers.

APPENDIX

- **Source code:** Provided in Apex classes and triggers

Code for Apex classes

```
public class AmountDistributionHandler {  
    public static void amountDist(list<Appointment__c> listApp){  
        list<Service_records__c> serList = new list <Service_records__c>();  
        for(Appointment__c app : listApp){  
            if(app.Maintenance_service__c == true && app.Repairs__c == true &&  
app.Replacement_Parts__c == true){  
                app.Service_Amount__c = 10000;  
            }  
            else if(app.Maintenance_service__c == true && app.Repairs__c == true){  
                app.Service_Amount__c = 5000;  
            }  
            else if(app.Maintenance_service__c == true && app.Replacement_Parts__c ==  
true){  
                app.Service_Amount__c = 8000;  
            }  
        }  
    }  
}
```

```

        }
        else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
            app.Service_Amount__c = 7000;
        }
        else if(app.Maintenance_service__c == true){
            app.Service_Amount__c = 2000;
        }
        else if(app.Repairs__c == true){
            app.Service_Amount__c = 3000;
        }
        else if(app.Replacement_Parts__c == true){
            app.Service_Amount__c = 5000;
        }
    }
}

```

Code for Apex triggers

```

trigger AmountDistribution on Appointment__c (before insert, before update) {

    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);

    }
}

```